INFORMATYKA
– MÓJ SPOSÓB NA POZNANIE I OPISANIE ŚWIATA

PROGRAM NAUCZANIA INFORMATYKI Z ELEMENTAMI PRZEDMIOTÓW MATEMATYCZNO-PRZYRODNICZYCH

Informatyka – poziom rozszerzony

Metody projektowania algorytmów

Paweł Perekietka

Człowiek - najlepsza inwestycja

KAPITAŁ LUDZKI
NARODOWA STRATEGIA SPÓJNOŚCI

Warszawska Wyższa Szkoła Informatyki

UNIA EUROPEJSKA
EUROPEJSKI FUNDUSZ SPOŁECZNY

Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego

Tytuł: ***Metody projektowania algorytmów***

Autor: ***Paweł Perekietka***

Redaktor merytoryczny: ***prof. dr hab. Maciej M. Sysło***

Materiał dydaktyczny opracowany w ramach projektu edukacyjnego
Informatyka – mój sposób na poznanie i opisanie świata.
Program nauczania informatyki z elementami przedmiotów
matematyczno-przyrodniczych

www.info-plus.wwsi.edu.pl

infoplus@wwsi.edu.pl

Wydawca: Warszawska Wyższa Szkoła Informatyki
ul. Lewartowskiego 17, 00-169 Warszawa
www.wwsi.edu.pl
rektorat@wwsi.edu.pl

Projekt graficzny: *Marzena Kamasa*

Warszawa 2013

Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego

SCENARIUSZ TEMATYCZNY

METODY PROJEKTOWANIA ALGORYTMÓW

INFORMATYKA – POZIOM ROZSZERZONY

OPRACOWANY W RAMACH PROJEKTU:

INFORMATYKA – MÓJ SPOSÓB NA POZNANIE I OPISANIE ŚWIATA.

PROGRAM NAUCZANIA INFORMATYKI

Z ELEMENTAMI PRZEDMIOTÓW MATEMATYCZNO-PRZYRODNICZYCH

Streszczenie

Wśród wymagań szczegółowych podstawy programowej wymienione są konkretne algorytmy, z którymi uczniowie powinni zapoznać się w czasie zajęć z informatyki rozszerzonej. Należy podkreślić, że zaznajomienie się z pewnym zasobem algorytmów nie jest celem samym w sobie – o wiele większą wartość edukacyjną ma poznanie przy okazji technik algorytmicznych, które znalazły zastosowanie przy projektowaniu algorytmu, rozwiązującego taki czy inny problem. Jak podkreślają autorzy jednego z poradników metodycznych: „Problemy i algorytmy, będące przedmiotem zajęć, służą ilustracji podstawowych technik rozwiązywania problemów, które mogą zostać zastosowane w wielu innych sytuacjach problemowych” [*Informatyka. Kształcenie w zakresie rozszerzonym*, (red.) M.M. Sysło, WSiP, Warszawa 2004, s. 71].

Co to jest technika algorytmiczna? Autorzy wspomnianej książki definiują ją w następujący sposób: „To ogólne podejście do rozwiązywania problemów, które dopiero w algorytmie ma swój precyzyjny zapis. Jako technikę algorytmiczną wyróżniamy te metody rozwiązywania, które (...) mają szersze zastosowanie, niż rozwiązanie jednego konkretnego problemu”. I dalej wymieniają pięć najważniejszych – zdaniem autorów – technik (metod) algorytmicznych, jakie informatycy-naukowcy wypracowali w ciągu kilkudziesięciu lat rozwoju algorytmiki: 1. Poszukiwanie i wybór elementów, 2. Iteracja i rekurencja, 3. Podejście „dziel i zwyciężaj”, 4. Metoda zachłanna oraz 5. Programowanie dynamiczne.

W nowej podstawie programowej *explicite* jest mowa o technikach algorytmicznych w następujących wymaganiach szczegółowych: 5.5 Uczeń posługuje się podstawowymi technikami algorytmicznymi, 5.8 Uczeń posługuje się metodą „dziel i zwyciężaj” w rozwiązywaniu problemów, 5.9 Uczeń stosuje rekurencję w prostych sytuacjach problemowych oraz 5.10 Uczeń stosuje podejście zachłanne w rozwiązywaniu problemów.

W ciągu ostatnich kilkunastu lat w wyniku prowadzonych badań naukowych i dyskusji dopracowana została bardziej kompletna systematyka (klasyfikacja) technik projektowania algorytmów (por. A. Levitin, *Introduction to the Design & Analysis of Algorithms*, 3rd edition, Pearson 2012). Dokonano przede wszystkim pogrupowania wcześniej już znanych technik w zależności od poziomu ich ogólności (uniwersalności). Podejściu „siłowemu” (ang. *brute force*), czyli prostolinijnej technice rozwiązywania problemów polegającej na wielokrot-

nym powtarzaniu prostej procedury (np. bezpośrednim sprawdzaniu wszystkich możliwości z określonego zbioru), dano prawo bytu w tej klasyfikacji (dostrzegając wartość dydaktyczną). Wyodrębniono też i scharakteryzowano inne techniki, co pozwoliło w nowej perspektywie spojrzeć na wiele algorytmów, które do tej pory wymykały się próbie klasyfikacji pod tym względem.

Techniki bardziej uniwersalne	**Techniki mniej uniwersalne**
podejście „siłowe" (ang. *brute force approach*)	metoda zachłanna
zmniejsz i zwyciężaj	programowanie dynamiczne
dziel i zwyciężaj	przeszukiwanie z nawrotami
przekształć i zwyciężaj	metoda podziału i ograniczeń

Co było podstawą uznania techniki za uniwersalną? Duże znaczenie miało bardzo konkretne kryterium: jej obecność w projektach istniejących algorytmów sortowania i porządkowania.

Wydaje się, że zapoznanie się z wynikami najnowszych badań w dziedzinie metod projektowania algorytmów komputerowych jest wskazane również na poziomie szkoły średniej. Do zilustrowania technik algorytmicznych, w pierwszym spotkaniu z nimi przez uczniów, warto posłużyć się łamigłówkami i zagadkami – w ostatnich latach wzbogacono klasyczny zestaw tzw. łamigłówek algorytmicznych (znanych od ponad 30 lat z podręczników akademickich czy publikacji popularnonaukowych). Jakie są zalety takiego podejścia? Anany Levitin, twórca nowej taksonomii technik algorytmicznych i autor zbioru „Algorithmic puzzles" pisze na temat łamigłówek tak:

- zmuszają one do myślenia o algorytmach na bardziej abstrakcyjnym poziomie (w oderwaniu od kontekstu programistycznego i związanych z nim szczegółów dotyczących języka programowania),
- ukazują techniki algorytmiczne jako techniki rozwiązywania problemów, które mogą być pomocne poza dziedziną informatyki,
- ich rozwiązywanie jest pomocne w rozwijaniu postawy twórczej i kształceniu umiejętności rozwiązywania problemów (w kontekście przygotowania do pracy zawodowej w dziedzinie informatyki),
- przyciągają uwagę (są bardziej atrakcyjne od typowych zadań), co prowadzi do wzrostu motywacji do pracy nad problemem, powiązanym z łamigłówką.

Lekcje algorytmiki i programowania nie mogą oczywiście pomijać wątku techniki projektowania klasycznych algorytmów. Zapoznanie się z klasyfikacją technik (metod) algorytmicznych na lekcjach z łamigłówkami powinno pozwolić uczniom w przyszłości na lepsze zrozumienie kolejnych przykładów zastosowania danej techniki (już dla klasycznych algorytmów), co bez wątpienia powinno też skutkować lepszym (pogłębionym) zrozumieniem samego przykładu (algorytmu).

W tym scenariuszu jest to ukazane na przykładzie różnych algorytmów dla problemu potęgowania. To temat ostatniej z cyklu lekcji. W podstawie programowej mówi się wprost tylko o jednym z algorytmów szybkiego potęgowania: związanym z zastosowaniem schematu Hornera (to bez wątpienia przykład zastosowania techniki przekształć i zwyciężaj). Okazuje się, że można wskazać co najmniej cztery różne algorytmy, ilustrujące różne techniki projektowania algorytmów.

Czas realizacji

8 x 45 minut

Podstawa programowa

Etap edukacyjny: IV, przedmiot: informatyka (poziom rozszerzony)

Cele kształcenia – wymagania ogólne

III. Rozwiązywanie problemów i podejmowanie decyzji z wykorzystaniem komputera, z zastosowaniem podejścia algorytmicznego.

Treści nauczania – wymagania szczegółowe

5.4 Uczeń dobiera efektywny algorytm do rozwiązania sytuacji problemowej i zapisuje go w wybranej notacji;

5.5. Uczeń posługuje się podstawowymi technikami algorytmicznymi;

5.8 Uczeń posługuje się metodą „dziel i zwyciężaj" w rozwiązywaniu problemów;

5.9 Uczeń stosuje rekurencję w prostych sytuacjach problemowych;

5.11c Uczeń opisuje zastosowania schematu Hornera: szybkie podnoszenie do potęgi.

Cele operacyjne

Uczeń:

- dostrzega wagę technik algorytmicznych dla projektowania algorytmów w informatyce,
- rozumie, na czym polega podejście „siłowe" (metoda bezpośrednia),
- rozumie ideę przeszukiwania z nawrotami i zapis przeszukiwania w postaci drzewa stanów,
- rozróżnia techniki zmniejsz i zwyciężaj, dziel i zwyciężaj oraz przekształć i zwyciężaj,
- przedstawia na przykładach ideę każdej z omawianych technik algorytmicznych,
- posługuje się poznanymi technikami do projektowania rozwiązań niezłożonych problemów,
- potrafi dostrzec różne techniki algorytmiczne w różnych projektach klasycznych algorytmów potęgowania.

Słowa kluczowe

technika (metoda) algorytmiczna, podejście „siłowe" (ang. *brute force*), permutacja, silnia, przeszukiwanie z nawrotami, drzewo stanów, wywołanie rekurencyjne, zmniejsz i zwyciężaj, dziel i zwyciężaj, przekształć i zwyciężaj, reprezentacja binarna liczby

Co przygotować

- Prezentacja „Kwadrat magiczny"
- Animacja „Czterech hetmanów na szachownicy"

Przebieg pierwszej lekcji

Wprowadzenie (5-10 minut)

Na początku zajęć nauczyciel przedstawia następujące zadanie-łamigłówkę:

Wypełnij pola kwadratowej tablicy o wymiarach 3 x 3 różnymi liczbami od 1 do 9 w taki sposób, aby suma liczb w każdym wierszu, w każdej kolumnie i na każdej głównej przekątnej była taka sama (tzw. suma magiczna).

?	?	?
?	?	?
?	?	?

Zadanie to ma służyć prezentacji podejścia „siłowego" do zadania kombinatorycznego, które polega na systematycznym (bezpośrednim) przeszukiwaniu całego zbioru potencjalnych rozwiązań (ang. *exhaustive search*), aż do momentu znalezienia rozwiązania. Trzeba zaznaczyć, że wielu informatyków-naukowców nie traktuje podejścia „siłowego" jako techniki projektowania w sensie ścisłym. Przynajmniej w przypadku problemów algorytmicznych, które nie należą to problemów trudnych obliczeniowo.

Nauczyciel stawia pytanie: „Na ile sposobów możemy wypełnić tabelę dziewięcioma liczbami?". Następnie nauczyciel powinien przedstawić rozumowanie (jeśli poprawna odpowiedź, tj. 9!, pojawiła się w odpowiedziach uczniów, to może powinien najpierw poprosić o „rozwinięcie" tej odpowiedzi) oparte na prostej zasadzie kombinatorycznej – zasadzie mnożenia, posługując się słowami podobnymi to następujących: „Jest dziewięć miejsc, w których możemy wstawić liczbę 1, osiem miejsc dla liczby 2 itd., zaś pozycja liczby 9 jest już jednoznacznie ustalona. Mamy więc $9 \times 8 \times ... \times 1 = 362\,880$ możliwości".

Prawdopodobnie w tym momencie uczniowie zwrócą uwagę na to, że autorzy łamigłówek (zagadek) zwykle nie oczekują od czytelników rozwiązań „siłowych". Bez wątpienia jest to prawda. Nauczyciel powinien stwierdzić, że w przypadku zadania tworzenia kwadratów magicznych rozwiązanie „siłowe" jest niepraktyczne również dla komputera. Dlaczego? (Okazuje się, że już dla $n = 5$ liczba permutacji jest równa $(52)! > 1025$. Sprawdzenie wszystkich ustawień wymaga ponad 30 tysięcy lat obliczeń dla komputera, który w ciągu jednej sekundy byłby w stanie przeprowadzić test tzw. sumy magicznej dla 10 bilionów, tj. 1013 różnych ustawień).

Zasadnicza część pierwszej lekcji (30-35 minut)

1. Nauczyciel pyta o pomysł bardziej efektywnej metody rozwiązania zadania (innej od rozpatrywania wszystkich permutacji). Wysłuchuje pomysły i sugestie uczniów. Następnie uruchamia prezentację „Kwadrat magiczny", która prezentuje rozwiązanie zawierające element techniki zwanej przeszukiwaniem z nawrotami lub nawracaniem (ang. *backtracking*). Prezentacja zawiera krótkie notatki (komentarze), które nauczyciel może wykorzystać przygotowując się do lekcji. Szczególną uwagę należy zwrócić na slajd 8, który ukazuje, na czym polega idea nawracania (nawrotu) – będzie ona ilustrowana kolejnym przykładem w czasie lekcji.

   Po zakończeniu prezentacji warto powiedzieć uczniom, że istnieją efektywne algorytmy tworzenia większych kwadratów magicznych (zwłaszcza dla n nieparzystych). Co ciekawe nie został jednak odkryty żaden wzór na liczbę kwadratów magicznych dla danego n.

2. W drugiej części lekcji nauczyciel przedstawia klasyczną łamigłówkę o rozmieszczeniu hetmanów (królowych) na szachownicy:

   Rozmieść czterech hetmanów na szachownicy 4 x 4 w taki sposób, aby żadne dwa nie atakowały się nawzajem.

Warto, by nauczyciel przygotował kilka folii (odpowiednich rozmiarów) z wyrysowanymi szachownicami, wzbogaconymi o współrzędne. Następnie należy podzielić klasę na 5- lub 6-osobowe grupy.

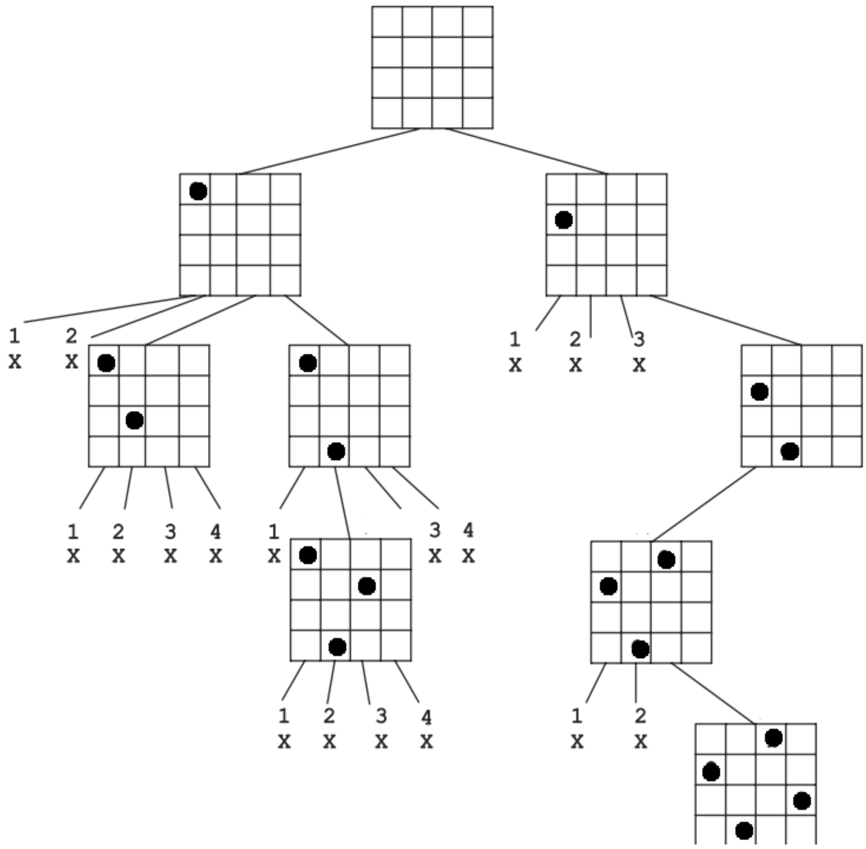
	1	2	3	4
1				
2				
3				
4				

Uczniowie powinni bez trudu rozwiązać zadanie – czterech (czworo) powinno wejść w „rolę" hetmanów. (Ta część lekcji może być prowadzona na korytarzu, jeśli wewnątrz brakuje miejsca). Różne grupy uczniów mogą uzyskać różne rozwiązania (są dwa!). Po znalezieniu rozwiązania uczeń „nie-hetman" powinien je zanotować na kartce. Następnie wybrani uczniowie zapisują rozwiązanie na tablicy.

Nauczyciel stawia pytanie: Czy potraficie odtworzyć proces poszukiwania rozwiązania? Uczniowie mogą uznać pytanie za mało istotne (komentując np. tak: „To proste zadanie!"). Wówczas nauczyciel pyta: Czy potrafilibyście „wytłumaczyć" sposób postępowania komputerowi?

Celem tej części lekcji jest ilustracja metody algorytmicznej zwanej przeszukiwaniem z nawrotami (nawracaniem), która stanowi usprawnienie techniki pełnego przeszukiwania. Korzysta się w niej z tego, że dla niektórych problemów już na etapie tworzenia elementów składowych potencjalnego rozwiązania możliwa jest jednoznaczna ocena braku rozwiązania dla podzbioru (często dużego) możliwych rozwiązań – pozwala to na wycofanie się ze „ślepej uliczki" tak szybko, jak to jest tylko możliwe.

Zwykle proces przeszukiwania (podejmowane decyzje) można zilustrować, używając tzw. drzewa stanów. W tym momencie nauczyciel powinien wyświetlić drzewo stanów dla problemu czterech hetmanów i przeanalizować krok po kroku z uczniami.



Powinien w szczególności zwrócić uwagę na zrozumienie idei „nawracania" (a więc i umiejscowienie „nawrotów" na powyższym diagramie – X oznacza, że nie można ustawić hetmana w odpowiednim wierszu, np. ciąg czterech X pod 1, 2, 3 i 4 oznacza „ślepą uliczkę" i potrzebę nawrotu).

Łamigłówka posiada drugie rozwiązanie (można je otrzymać jako przekształcenie planszy w symetrii osiowej). Uczniowie powinni podjąć próbę uzupełnienia drzewa stanów.

Podsumowanie lekcji (5-10 minut)

W ramach podsumowania nauczyciel w rozmowie z uczniami powinien oszacować wielkość usprawnienia, jakie w przypadku zadania czterech hetmanów daje zastosowanie przeszukiwania z nawrotami. Jeśli przy podejściu „siłowym" analizowalibyśmy tylko te ustawienia, w których każdy hetman jest w innej kolumnie, to liczba różnych ustawień hetmanów jest równa 44 = 256.

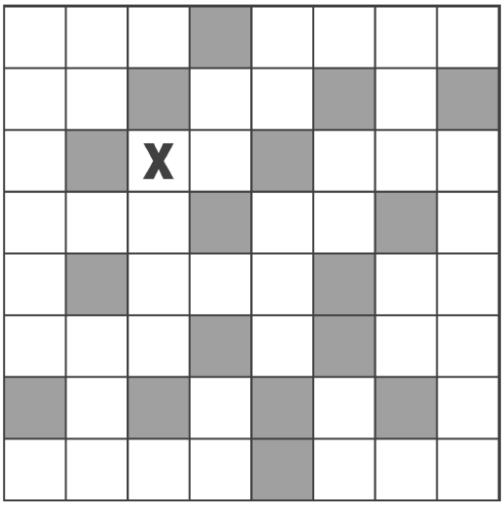
Przy okazji warto uczniom zwrócić uwagę na to, że istnieją znacznie bardziej efektywne algorytmy rozwiązywania problemu hetmanów i to dla dowolnego *n*.

Ciekawe jest to, że fakt istnienia bardziej efektywnych algorytmów był przez wiele lat ignorowany (czasem wręcz nieznany) przez informatyków – wydaje się, że walor edukacyjny rozwiązywania łamigłówki przez przeszukiwanie z nawrotami sprawił, że powstało fałszywe wrażenie, że problem jest trudny obliczeniowo. Trzeba podkreślić, że przewaga metody przeszukiwania z nawrotami jest taka, że pozwala na znalezienie wszystkich rozwiązań – np. dla wersji 8 x 8 jest ich 92 (w tym 12 różnych z dokładnością do symetrii).

Nauczyciel powinien na zakończenie podkreślić, że metoda przeszukiwania z nawrotami zazwyczaj nie jest metodą bardzo efektywną, ale czasami jedyną skuteczną. I poinformować, że celem następnych lekcji będzie poznanie bardziej innych metod projektowania algorytmów.

Zadanie domowe 1

Znajdujesz się w labiryncie w miejscu oznaczonym jako X. Odszukaj miejsce najbardziej oddalone od X, tzn. takie, które wymaga wykonania największej liczby ruchów, aby do niego dotrzeć. Jaka jest ta liczba ruchów?



Uwaga: Zakładamy, że w jednym ruchu możesz przesunąć się o dowolną, możliwą w danym miejscu, liczbę pól w kierunku poziomym lub pionowym.

Wskazówka: Zacznij od oznaczenia liczbą 1 miejsc, do których możesz dotrzeć w jednym ruchu. Następnie oznacz liczbą 2 te miejsca, do których dotrzesz w dwóch ruchach. Itd.

Odpowiedź: 9

Zadanie domowe 2

Dany jest zbiór {5, 10, 12, 13, 15, 18}. Znajdź wszystkie podzbiory tego zbioru, których elementy dają w sumie 30. Proces przeszukiwania (wyznaczania podzbioru) przedstaw, używając tzw. drzewa stanów.

Wskazówka: W lewym poddrzewie drzewa stanów znajdują się te rozwiązania (o ile istnieją), do których należy liczba 5, a w prawym – te, do których liczba 5 nie należy.

Odpowiedź: {5, 10, 15}, {5, 12, 13} oraz {12, 18}.

Przebieg kolejnych lekcji

Lekcje 2-7 powinny być prowadzone metodą aktywizującą o nazwie „grupy eksperckie" (tzw. metoda układanki). Jest ona przykładem efektywnej strategii uczenia się, w której przyswajana przez ucznia wiedza jest wykorzystywana do nauczania innych.

Metoda wymaga czasu i zachowania dyscypliny przy organizacji pracy tą metodą. Uczniowie powinni odpowiednio wcześnie poznać zasad pracy i etapy, które są następujące:

1. Nauczyciel dzieli uczniów na tyle grup roboczych, ile jest wyodrębnionych tematów:
   - Zmniejsz i zwyciężaj (ang. *decrease-and-conquer*),
   - Dziel i zwyciężaj (ang. *divide-and-conquer*),
   - Przekształć i zwyciężaj (ang. *transform-and-conquer*).

   Każdy uczeń otrzymuje materiał (patrz: materiały pomocnicze), który zawiera wprowadzenie w temat oraz zadania sprawdzające, czy opanował zagadnienie w wystarczającym stopniu.
2. Uczniowie opracowują materiał indywidualnie (w czasie pierwszej lekcji oraz w domu). Rozwiązują zadania przygotowane przez nauczyciela. W razie potrzeby zgłaszają nauczycielowi wątpliwości (np. dotyczące jednoznaczności sformułowań w tekście).
3. W czasie drugiej lekcji w grupach roboczych odbywa się runda ekspertów. Uczniowie wyjaśniają sobie nawzajem wątpliwości i zaistniałe problemy. Porównują rozwiązania zadań. Wszystko po to, aby każdy z ekspertów poczuł się bardziej pewnie w danym temacie.
4. W czasie drugiej części drugiej lekcji członkowie grup roboczych powinni zaplanować, jak przekazać zdobytą wiedzę innym.
5. W czasie kolejnych trzech lekcji w grupach prezentacyjnych (utworzonych przez wzięcie po jednej osobie z każdej grupy roboczej) odbywa się runda lekcji. Każdy uczeń prezentuje na swój temat pozostałym uczniom z grupy (w każdej grupie omawiane są równolegle te same tematy). Po prezentacji uczniowie rozwiązują zadania w grupach. Następnie rozwiązanie zadania prezentowane jest przez jedną w grup na forum całej klasy. Następuje dyskusja. Nauczyciel wyjaśnia wątpliwości i w razie potrzeby odpowiednio koryguje przedstawione rozwiązanie. Powinien zwrócić szczególną uwagę na subtelną różnicę między technikami dziel i zwyciężaj oraz zmniejsz i zwyciężaj (w wersji „zmniejsz dwa razy").
6. W domu (lub jeszcze podczas lekcji) uczniowie powinni zmierzyć się z zadaniami „bez nagłówków", tzn. pozbawionych wskazówki co do metody algorytmicznej. Praca uczniów powinna być oceniana. Na ostatniej lekcji powinna nastąpić prezentacja rozwiązań.

   O takiej czy innej formie pracy kontrolnej, kończącej cykl lekcji, nauczyciel powinien wyraźnie poinformować uczniów na początku cyklu lekcji, aby zmobilizować ich do solidnej pracy w zespołach roboczych i grupach prezentacyjnych. Uczniów wyróżniających się wcześniej podczas sesji prezentacyjnej nauczyciel może zwolnić z obowiązku pisania pracy kontrolnej.

Zadania domowe 3

(Gilotyna)

Papier o rozmiarach 10 x 100 ma zostać pocięty na 100 kawałków o rozmiarach 10 x 1. Jaka jest najmniejsza liczba potrzebnych cięć, jeśli gilotyna pozwala na cięcie nawet 50 kawałków papieru naraz?

(Kamienie)

Dwóch graczy gra w następującą grę. Zaczynają od stosu składającego się z 50 kamieni. Wykonują ruchy na przemian. W każdym kroku każdy z graczy musi usunąć co najmniej jeden i co najwyżej p kamieni. Gracz, który usuwa ostatni kamień (ostatnie kamienie) przegrywa.

a) Zaprojektuj algorytm dla $p = 6$, który pozwoli rozpoczynającemu grę na zwycięstwo niezależnie od strategii zastosowanej przez przeciwnika.

b) Dla jakiego p nie istnieje strategia wygrywająca dla rozpoczynającego grę?

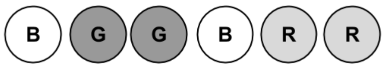
(Idol raz jeszcze)

Załóżmy, że nie mamy pewności co do obecności idola na przyjęciu. Jak sprawdzić, czy osoba niewyeliminowana z użyciem procedury, przedstawionej w czasie lekcji, jest rzeczywiście idolem?

(Żetony RGB)

Masz trzy zestawy żetonów (czerwone, zielone i niebieskie), ustawione w jednej linii. Chcesz je uporządkować tak, aby czerwone (R) żetony były po lewej stronie, zielone (G) w środku i niebieskie (B) po prawej stronie.

Porządkowanie polega na zamianie (miejscami) dwóch żetonów. Możesz zamienić dwa żetony miejscami tylko wtedy, gdy pomiędzy nimi masz dokładnie tylko jeden inny żeton.



Oznacza to na przykład, że porządkowanie układu BGGBRR może wyglądać tak:

BGGBRR -> **G**G**B**BRR -> GG**R**B**B**R -> GG**R**RBB -> G**R**R**G**BB -> **R**R**G**GBB

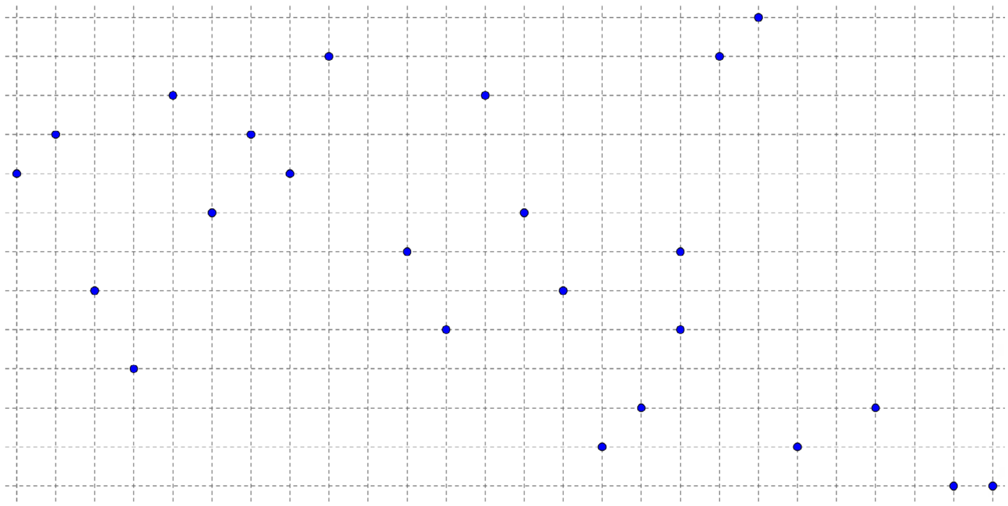
Ile potrzeba zamian, dla uporządkowania układu BGBGGRGRRBRB?

(Nowa figura szachowa)

Wyobraź sobie nową figurę szachową o nazwie Książę, którą możesz wykonywać ruchy: o jedno pole w prawo, o jedno pole w dół lub o jedno pole na ukos do góry w lewo. Jak przejść Księciem, wyruszając z jakiegokolwiek pola, przez każde pole szachownicy tylko raz?

(Ośmiokąty)

Na kartce w kratkę narysowano 24 punkty. Narysuj trzy ośmiokąty proste w taki sposób, że żadne dwa z nich nie będą mieć punktów wspólnych.



Uwaga: Wielokąt prosty to wielokąt, którego boki tworzą zamkniętą łamaną, a dwa jego boki mają punkt wspólny, tylko gdy są sąsiednimi bokami.

(Tromino raz jeszcze)

Wypełnić szachownicę o rozmiarach 12 x 12 sześciokątami L-tromino.

(Domino)

Wypełnić klockami domino szachownicę rozmiarów *n* x *n*, z której usunięto dwa pola.

(Odważniki raz jeszcze)

Zaprojektuj zestaw takich dziesięciu odważników do wagi szalkowej, który umożliwi ważenie jak największej liczby przedmiotów z dokładnością do 1 kg.

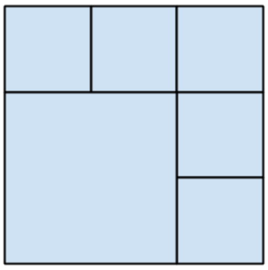
Jaki może być największy ciężar ważonego przedmiotu?

Zakładamy, że odważniki można kłaść na obu szalkach wagi.

(Podział kwadratu)

Uzasadnij, że kwadrat można podzielić na n mniejszych kwadratów dla każdego $n > 5$.

Rysunek poniżej ilustruje podział dla $n = 6$.



(Odgadywanie raz jeszcze)

Algorytm dla wersji gry *Odgadywanie*, gdy zamiast 1000 mamy liczbę n jest następujący:

- Przyjmij $p = 1$ i $k = n$.
- Dopóki $p \neq k$, tzn. dopóki nie zredukujesz zbioru zawierającego poszukiwaną liczbę do zbioru jednoelementowego, postępuj w następujący sposób:
  - Zadaj pytanie „Czy to liczba większa od $\left\lfloor \frac{p+k}{2} \right\rfloor^{*}$?"
  - Jeśli odpowiedź na pytanie brzmi „Nie", to przyjmij $k = \left\lfloor \frac{p+k}{2} \right\rfloor$
  - W przeciwnym przypadku przyjmij, że $p = \left\lfloor \frac{p+k}{2} \right\rfloor + 1$.
- Poszukiwaną liczbą jest p.

(*) Zapis $\lfloor x \rfloor$ (czyt. „podłoga z x") oznacza największą liczbę całkowitą niewiększą niż x.

Zapisz algorytm, używając języka programowania lub formuł arkusza kalkulacyjnego.

Przebieg ostatniej lekcji

Wprowadzenie

Dwie ostatnie lekcje tego cyklu lekcji powinny służyć ukazaniu obecności poznanych przez uczniów technik algorytmicznych w projektach konkretnych klasycznych algorytmów.

Przykładem są algorytmy porządkowania. W literaturze (np. książkach A. Levitina) można znaleźć takie zestawienie:

podejście „siłowe"	porządkowanie przez wybór porządkowanie bąbelkowe
zmniejsz i zwyciężaj	porządkowanie przez wstawianie porządkowanie przez wybór
dziel i zwyciężaj	porządkowanie przez scalanie porządkowanie szybkie
przekształć i zwyciężaj	porządkowanie przez kopcowanie

Wszystkie poza ostatnim to tematy z podstawy programowej. Warto w czasie realizacji w klasie poszczególnych tematów dotyczących algorytmów porządkowania, wracać do tego zestawienia tak, aby zrozumienie zagadnienia technik projektowania algorytmów nie było w przekonaniu uczniów zagadnieniem teoretycznym, odizolowanym od reszty zagadnień.

Wydaje się, że szczególny walor edukacyjny w szkole ponadgimnazjalnej może mieć inny przykład – algorytmami potęgowania (podnoszenia do potęgi).

Uwaga: Lekcja powinna być realizowana bezpośrednio po wcześniejszych lekcjach. Ma ona charakter niezbędnego – jak się wydaje – dopowiedzenia. Jej celem jest również doskonalenie umiejętności analizy działania algorytmów i zapisywania algorytmów w wybranej notacji.

Szczegółową lekcję związaną z algorytmem potęgowania, wymienionym w podstawie programowej, warto zrealizować przy okazji omawiania algorytmu RSA (algorytm szybkiego podnoszenia do potęgi jest elementem komputerowej realizacji RSA) lub dopiero wtedy, gdy uczniowie będą mieć odpowiednie kompetencje z zakresu programowania (m.in. konstrukcje iteracyjne) – wtedy będą mogli zająć się komputerową realizacją algorytmów.

Uczniów zainteresowanych problemem algorytmów szybkiego podnoszenia do kwadratu warto odesłać do szczegółowych opisów w książkach popularnonaukowych autorstwa prof. Macieja M. Sysły.

Przebieg lekcji

1. Nauczyciel rozpoczyna lekcję od przypomnienia nazw (i krótkiej charakterystyki) uniwersalnych technik algorytmicznych: podejście „siłowe", zmniejsz i zwyciężaj, dziel i zwyciężaj oraz przekształć i zwyciężaj. Uczniowie mogą podać przykłady łamigłówek, które im się kojarzą z każdą ze strategii.
2. Następnie nauczyciel formułuje problem podnoszenia do potęgi:

   Dla liczb naturalnych *a*, *n* (a > 0 i n>= 0) chcemy wyliczyć wartość potęgi *an*.

   I prosi uczniów o uzupełnienie pseudokodu (wynik musi być poprawny dla *n* = 0) dobrze im znanego tzw. algorytmu naiwnego, który jest przykładem zastosowania podejścia „siłowego" (tj. bezpośredniego korzystania z definicji $a^n = \underbrace{a \cdot \ldots \cdot a}_{n \text{ razy}}$):

```
czytaj(a, n)
b <- ...
k <- ...
dopóki k <=...
    b <- b * a
    k <- k +...
pisz(b)
```

a następnie realizację algorytmu na kartce dla $a = 2$ i $n = 9$, tj. zapis wartości, jakie przyjmują b i k. Celem jest również szukanie odpowiedzi na pytanie o liczbę mnożeń wykonywanych przez algorytm.

Jeśli notacja pseudokodu jest uczniom nieznana, ćwiczenie może polegać na uzupełnieniu algorytmu w postaci listy kroków lub schematu blokowego algorytmu. Oczywiście, jeśli tylko umiejętności uczniów na to pozwalają, to nauczyciel powinien polecić IM zaprogramowanie algorytmu.

Omawianie tego przykładu nauczyciel kończy zapisaniem algorytmu naiwnego w trochę inny sposób:

```
czytaj(a, n)
b <- 1
k <- n
dopóki k >= 1
    b <- b * a
    k <- k – 1
pisz(b)
```

Jaki jest cel wprowadzania tej zmiany, będzie jasne dla uczniów w dalszej części lekcji.

3. Nauczyciel stawia pytanie: „Czy można w bardziej efektywny sposób obliczyć *an*?"

   Warto ukonkretnić pytanie: „Jak obliczyć *a16*, wykonując mniej niż 15 mnożeń?".

   Oczekujemy odpowiedzi: *a16* = (((*a*2)2)2)2

   W tym momencie powinniśmy zapytać o strategię wykorzystywaną przy realizacji tego pomysłu? To przykład techniki zmniejsz i zwyciężaj.

   - Zapisujemy na tablicy formułę, która jest kluczowa w tym rozwiązaniu:

$$a^n = \begin{cases} (a^{n/2})^2 & \text{dla } n \text{ dodatnych parzystych} \\ \left(a^{(n-1)/2}\right)^2 \cdot a & \text{dla } n \text{ nieparzystych} \\ 1 & \text{dla } n = 0 \end{cases}$$

   - Proponujemy uczniom „realizację" na kartce powyższego algorytmu dla a = 2 i n = 9.

   Podajemy go uczniom w postaci pseudokodu. (Możemy oczywiście posłużyć się też schematem blokowym.)

```
czytaj(a, n)
b <- 1
k <- n
c <- a
dopóki k >= 1
```

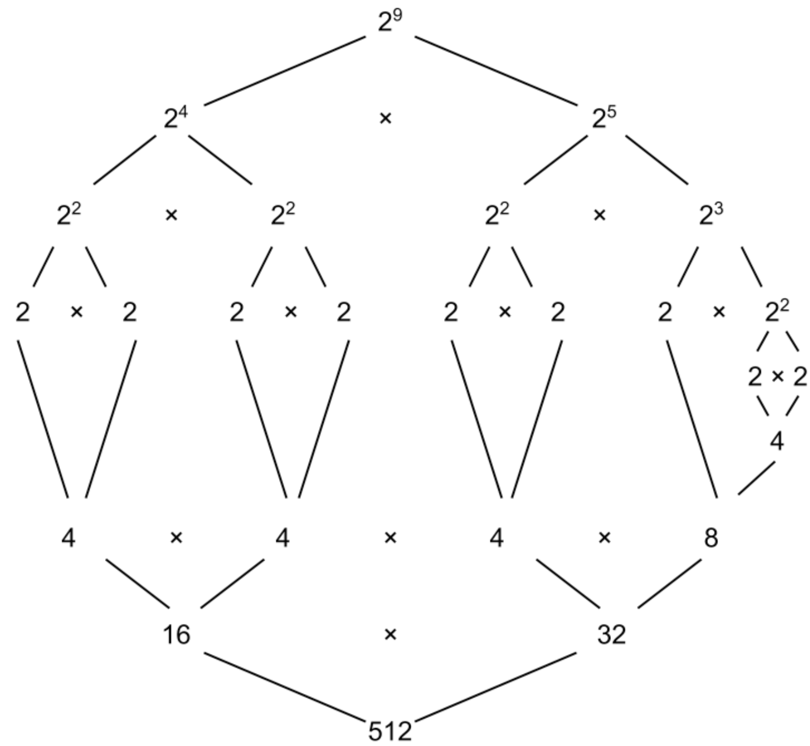
jeżeli (k **mod** 2) = 0
to wtedy
c <- c * c
k <- k div 2
w przeciwnym wypadku
b <- b * a
k <- k – 1
pisz(b)

Celem ćwiczenia jest również szukanie odpowiedzi na pytanie o liczbę mnożeń wykonywanych przez algorytm. Oczywiście, jeśli tylko umiejętności uczniów na to pozwalają, to nauczyciel powinien polecić uczniom zaprogramowanie algorytmu.

4. Ostatnim przykładem algorytmu będzie przykład, w którym zastosowanie znajduje technika dziel i zwyciężaj. Istotę algorytmu można opisać formułą:

$$a^1 = a$$
$$a^n = a^{\lfloor n/2 \rfloor} \cdot a^{\lceil n/2 \rceil} \text{ dla } n > 1$$

Schemat działania algorytmu dla $a = 2$ i $n = 9$ można przedstawić w postaci tzw. drzewa wywołań rekurencyjnych:



Również w tym przypadku uczniowie powinni policzyć liczbę wykonywanych mnożeń.

Przy okazji warto zwrócić uwagę uczniów na to, że ze względu na wielokrotne odwołania rekurencyjne komputerowa realizacja tego algorytmu nie jest godna polecenia.

5. Informujemy uczniów o zadaniu domowym (zadanie domowe 4). Jego istotą jest poznanie szybkiego podnoszenia do potęgi na przykładzie algorytmu, w którym korzysta się reprezentacji binarnej wykładnika i stosowany jest schemat Hornera. To przykład zastosowania strategii przekształć i zwyciężaj (następuje zmiana sposobu zapisu wykładnika potęgi – metoda ta wykorzystuje pośrednio dwójkową reprezentację wykładnika).

   Również w tym przypadku uczniowie powinni policzyć liczbę wykonywanych mnożeń.
6. Na zakończenie lekcji warto podkreślić, że problem szybkich algorytmów potęgowania jest problemem otwartym (w tym sensie, że nadal nie jest znany ogólny algorytm, który dla każdego wykładnika wykonywałby najmniejszą możliwą liczbę działań).

Ocenianie

Nauczyciel może oceniać osiągnięcia uczniów na podstawie obserwacji ich pracy i zaangażowania na lekcji oraz na podstawie prac przygotowanych w ramach zadania domowego.

Dostępne pliki

1. Prezentacja
2. Zadania 1-4
3. Animacja
4. Test
5. Materiały pomocnicze 1, 2 i 3 (załączniki A, B, C)

Materiały źródłowe

Informacje na temat metody przeszukiwania z nawrotami (m.in. o problemie rozmieszczenia hetmanów na szachownicy, poszukiwaniu wyjścia z labiryntu) oraz o problemie idola można przeczytać w książkach *Piramidy, szyszki i inne konstrukcje algorytmiczne* oraz *Algorytmy* prof. Macieja M. Sysły.

W obu wspomnianych książkach jest też mowa o algorytmach szybkiego podnoszenia do potęgi. Są tam m.in. ciekawostki historyczne (niektóre algorytmy były znane w starożytności).

Schematy blokowe algorytmów potęgowania można odnaleźć w wielu książkach, np. w podręczniku: E. Palka, *Elementy algorytmiki dla początkujących*, Poznań 2012.

Większość łamigłówek przedstawionych w scenariuszu pochodzi z książki: A. Levitin, M. Levitin, *Algorithmic puzzles*, Oxford University Press. 2011 oraz D. Clark, *The Australian Informatics Competition 2005-2010*. AMT Publishing 2011.

Załącznik A
(materiały pomocnicze 1)

Zmniejsz i zwyciężaj (ang. *decrease-and-conquer*)

Technika projektowania algorytmów zwana zmniejsz i zwyciężaj opiera się dostrzeżeniu i wykorzystaniu związku między rozwiązaniem pierwotnego problemu a rozwiązaniem tego samego problemu dla mniejszej liczby danych wejściowych.

Przykład 1 (Idol)

Na przyjęciu wśród 24 osób jest obecny idol, tj. osoba, które jest znana wszystkim, ale nie zna nikogo. Jak wykryć idola, stawiając uczestnikom przyjęcia pytania typu „Znasz tę osobę?".

a) Zaproponuj rozwiązanie „siłowe".

b) Zadanie w sposób efektywny można rozwiązać stawiając tylko 23 pytania. Jak to zrobić?

Rozwiązanie:

Sposób postępowania (algorytm) dla zbioru *n* osób jest następujący:

- Wybieramy dwie osoby (nazwijmy je A i B) i pytamy osobę A „Znasz osobę B?".
- Jeśli A zna B, to usuwamy osobę A z grona kandydatów na idola. W przeciwnym przypadku usuwamy z tego zbioru osobę B.
- Następnie powtarzamy powyższe kroki dla zbioru $n - 1$ osób, dopóki $n > 1$.

Projekt algorytmu to przykład zastosowania techniki typu zmniejsz o 1 i zwyciężaj. Zauważ, że zawiera on tzw. wywołanie rekurencyjne („powtarzamy powyższe kroki").

c) Zastosuje algorytm dla przykładu z inną liczbą osób. Ile pytań trzeba postawić?

Zadanie 1 (Przeprawa harcerzy)

Hufiec 25 harcerzy musi przeprawić się przez szeroką i głęboką rzekę. Harcerze spostrzegli dwóch chłopców w łódce, bawiących się przy brzegu rzeki. Łódka jest na tyle mała, że może się w niej zmieścić dwóch chłopców, ale tylko jeden harcerz. Jak mogą się przeprawić przez rzekę harcerze, jeśli łódka po przeprawieniu się harcerzy ma zostać zwrócona chłopcom?

a) Zastosuj algorytm wykorzystujący m.in. technikę typu zmniejsz o 1 i zwyciężaj.

b) Ile razy podczas przeprawy łódka będzie przepływać z jednego brzegu na drugi?

Przykład 2 (Odgadywanie)

Piotr wybrał liczbę naturalną nie większą niż tysiąc. Ania chce ustalić wartość tej liczby zadając jak najmniejszą liczbę pytań, na które Piotr może odpowiadać tylko „Tak" lub „Nie". Jak powinna postępować Ania, aby pytań nie było więcej niż 10?

Rozwiązanie:

Najlepszy możliwy sposób postępowania dla Ani polega na stawianiu takich pytań, których skutkiem jest zmniejszenie o połowę zbioru zawierającego poszukiwaną liczbę. Oznacza to, że pierwsze pytanie powinno brzmieć tak: „Czy to liczba większa od 500?"

Jak powinno brzmieć następne pytanie w zależności od odpowiedzi na pierwsze pytanie?

Załóżmy, że Piotr wybrał liczbę 88, a Ania stosuje optymalną strategię.

Przeanalizujmy przebieg gry, zapisując w tabeli kolejno:

- Numer pytania.
- Liczbę p, tj. najmniejszą (na danym etapie) liczbę, która może być poszukiwaną liczbą.
- Liczbę k, tj. największą (na danym etapie) liczbę, która może być poszukiwaną liczbą.
- Liczbę l elementów zbioru, przeszukiwanego na danym etapie.
- Pytania Ani i odpowiedzi Piotra.

Lp.	*p*	*k*	*l*	Ania	Piotr
1	1	1 000	1000	> 500?	Nie
2	1	500	500	> 250?	Nie
3	1	250	250	> 125?	Nie
4	1	125	125	> 63?	Tak
5	64	125	62	> 94	Nie
6	64	94	31	> 79	Tak
7	79	94	16	> 86	Tak
8	86	94	9	> 90	Nie
9	86	90	5	> 88	Nie
10	86	88	3	> 87	Tak

Po zadaniu dziesięciu pytań Ania już wie, że poszukiwaną liczbą jest 88.

Można powiedzieć, że Ania zastosowała technikę typu „zmniejsz dwa razy i zwyciężaj”. Powyższa tabela dobrze to ukazuje: z każdym kolejnym pytaniem rozmiar zadania (zakres liczb, wśród których znajduje się poszukiwana liczba) jest mniej więcej dwa razy mniejszy.

a) Zapisz w tabeli „historię” gry dla innej liczby.

b) Jaka będzie wystarczająca liczba pytań w grze *Zgadywanie* dla *n* równego 1 000 000?

Zadanie 2 (Monety)

Mamy osiem identycznie wyglądających monet. Wiemy, że jedna z nich jest lżejsza od pozostałych. Ilu ważeń na wadze szalkowej potrzeba wykonać, aby znaleźć monetę lżejszą, przy założeniu, że na szalki wagi można wkładać dowolną liczbę monet?

a) Zastosuj algorytm wykorzystujący m.in. technikę typu zmniejsz dwa razy i zwyciężaj. Ile ważeń jest potrzebnych?

b) Jak wykryć fałszywą monetę, używając tylko dwóch ważeń?

Jak nazwać technikę, zastosowaną tym razem do zmniejszenia rozmiaru zadania?

c) Czy rozwiązania mają charakter ogólny?

Czy można je zastosować do dowolnej liczby monet?

Do wykonania zadań wskazane jest użycie wagi szalkowej. Za odważniki mogą służyć pudełka zapałek.

Załącznik B
(materiały pomocnicze 2)

Dziel i zwyciężaj (ang. *divide-and-conquer*)

Technika algorytmiczna dziel i zwyciężaj polega na podziale oryginalnego problemu na pewną liczbę podproblemów (zwykle różniących się od oryginalnego tylko mniejszą liczbą danych wejściowych), ich rozwiązaniu oraz połączeniu tych rozwiązań w celu określenia rozwiązania pierwotnego zadania. Ta metoda projektowania algorytmów jest podstawą wielu efektywnych algorytmów w informatyce.

Przykład 1 (Żetony na szachownicy)

Rozmieść 16 żetonów na szachownicy 8 x 8 w taki sposób, aby w każdej kolumnie, w każdym wierszu i na obu przekątnych nie było ich więcej niż dwa.

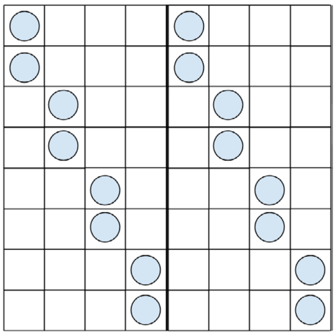
a) Rozwiąż zadanie, dzieląc zawarty w nim problem, w odpowiedni sposób na dwa podproblemy, które łatwiej rozwiązać.

Rozwiązanie:

Ponieważ liczba żetonów jest dwa razy większa od liczby kolumn i wierszy, więc liczba żetonów w każdej kolumnie i każdym wierszu musi być równa 2.

Powstaje pytanie: Czy można rozmieścić żetony w identyczny sposób w pierwszych czterech kolumnach i czterech następnych kolumnach?

Okazuje się, że dość łatwo znaleźć takie ustawienie, co pokazane jest na rysunku:



Projekt algorytmu to przykład zastosowania techniki typu dziel i zwyciężaj: problem został podzielony na dwa mniejsze, które dość łatwo rozwiązać.

b) Jak zmodyfikować rozwiązanie pierwotnego zadania, aby uzyskać rozwiązanie dla przypadku 18 żetonów na szachownicy 9 x 9?

Zadanie 1 (Najcięższy i najlżejszy)

Mamy 16 identycznie wyglądających odważników. Ilu ważeń na wadze szalkowej potrzeba w celu wykrycia najlżejszego i najcięższego odważnika, przy założeniu, że na szalki wagi można wkładać tylko po jednym odważniku?

a) Zastosuj algorytm wykorzystujący m.in. technikę typu dziel i zwyciężaj.

b) Jak zmodyfikować rozwiązanie pierwotnego zadania, aby uzyskać rozwiązanie dla przypadku 17 odważników?

Do wykonania zadań wskazane jest użycie wagi szalkowej. Za odważniki mogą służyć pudełka zapałek.

Przykład 2 (Tromino)

Z szachownicy 8 x 8 usunięto jedno z pól (dowolne). Jak wypełnić pozostałe 63 jej pola sześciokątami L-tromino, które tworzą trzy kwadratów o rozmiarze pola szachownicy?

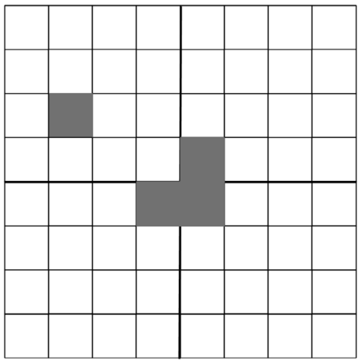
a) Zaproponuj rozwiązanie „siłowe". Użyj wielokątów L-tromino z papieru. Ile ich trzeba? Stosujesz metodę prób i błędów? Można w prosty sposób zastosować tę metodę do szachownicy 16 x 16, 32 x 32 itd.?

b) Zadanie można rozwiązać, stosując uporządkowaną procedurę. Jak to zrobić?

Rozwiązanie:

Sposób postępowania (algorytm) dla szachownicy o liczbie pól 2*n* jest następujący:

- Podziel szachownicę na cztery szachownice o dwa razy mniejszej liczbie pól.
- Umieść jedno L-tromino na środku szachownicy tak, jak to jest ukazane na rysunku:



- Jeśli liczba pól mniejszych szachownic jest większa od czterech, to powtórz powyższe kroki dla każdej z nich.

Projekt algorytmu to przykład zastosowania techniki typu dziel i zwyciężaj. Zauważ, że zawiera cztery tzw. wywołania rekurencyjne (powtórz powyższe kroki).

c) Zastosuj algorytm dla pierwotnego zadania, usuwając pole w innym miejscu szachownicy.

Zadanie 2 (Najbliższa para punktów)

Narysowano pewną liczbę punktów (ich odcięte są różne). Znajdź parę najbliższych punktów.

a) Zaproponuj rozwiązanie „siłowe".

b) Zaproponuj algorytm wykorzystujący m.in. technikę typu zmniejsz dwa razy i zwyciężaj, rozwiązujący problem dla dowolnej liczby punktów.

Załącznik C
(materiały pomocnicze 3)

Przekształć i zwyciężaj (ang. *transform-and-conquer*)

Technika przekształć i zwyciężaj to podejście do rozwiązywania problemu algorytmicznego, które składa się z dwóch etapów: najpierw problem jest przekształcany na inny (np. przez zmianę reprezentacji danych wejściowych), równoważny mu, który z jakiegoś powodu daje większe nadzieje na jego rozwiązanie; drugi etap to właśnie znajdowanie rozwiązania drugiego problemu, które jest jednocześnie rozwiązaniem pierwotnego problemu.

Przykład 1 (Anagramy)

Słowa tuba i buta czy atol i lato są anagramami, tzn. składają się z tych samych liter. Znajdź anagramy wśród słów:

abo	aft	akr	akt	alb	ale	ano	bal	boa	dal	dla	fag	gaf	kat	kra	ona	rak	taf	Tak

a) Zaproponuj rozwiązanie „siłowe".

b) W I etapie dokonaj zmiany reprezentacji (sposobu przedstawienia) danych wejściowych.

Rozwiązanie:

Sposób postępowania (algorytm) jest następujący:

- Dodajemy każdemu ze słów „sygnaturę", która ma postać uporządkowanego jak w alfabecie ciągu liter tworzących słowo (to etap zmiany reprezentacji).

abo	aft	akr	akt	alb	ale	ano	bal	boa	dal	dla	fag	gaf	kat	kra	ona	rak	taf	tak
abo	*aft*	*akr*	*akt*	*abl*	*ael*	*ano*	*abl*	*abo*	*adl*	*adl*	*afg*	*afg*	*akt*	*akr*	*ano*	akr	aft	akt

- Następnie porządkujemy w porządku alfabetycznym „sygnatury" i uzyskujemy rozwiązanie zadania – anagramy znajdują się obok siebie.

alb	bal	abo	boa	dal	dla	ale	fag	gaf	aft	taf	akr	kra	rak	akt	kat	tak	ano	Ona
abl	*abl*	*abo*	*abo*	*adl*	*adl*	*ael*	*afg*	*afg*	*aft*	*aft*	*akr*	*akr*	*akr*	*akt*	*akt*	*akt*	*ano*	*Ano*

Projekt algorytmu to przykład zastosowania techniki typu przekształć i zwyciężaj. Zastosowana została zmiana reprezentacji, tj. sposób przedstawienia danych wejściowych, poprzez uporządkowanie liter.

Należy podkreślić, że w przypadku większej liczby danych wejściowych trzeba by posłużyć się efektywnym algorytmem porządkowania.

Zadanie 1 (Układ nierówności)

Zastąp litery liczbami 2, 3, 1, 5, 8, 6, 11, 13, 10,

$a < b > c < d < e > f < g < h > i$,

tak, aby nierówności $a < b$, $b > c$, $c < d$ itd. były *prawdziwe:*

Wskazówka:

Dla liczb 2, 5, 1 i 0 i układu nierówności $a < b > c < d$ *rozwiązaniem jest np.* $0 < 5 > 1 < 2$, *gdyż* $0 < 5$, $5 > 1$ i $1 < 2$.

a) Przeanalizuj dokładnie wskazówkę.

b) Zastosuj algorytm wykorzystujący m.in. technikę typu przekształć i zwyciężaj.

Przykład 2 (Koperty)

Masz tysiąc banknotów o nominale 1$. Jak rozmieścić banknoty w 10 kopertach, aby przy pomocy pewnej kombinacji tych kopert dało się uzyskać każdą kwotę między 1$ i 1000$?

a) Zaproponuj rozwiązanie „siłowe".

b) Zauważ, że 210>1000. Jak rozwiązać zadanie, stosując własności numeracji dwójkowej?

Rozwiązanie:

Do zmiany reprezentacji (sposobu przedstawiania) danych wejściowych wykorzystamy reprezentację binarną liczb, a dalej skorzystamy z tego, że 9 bitów wystarczy do zapisania każdej liczby mniejszej niż 29 = 512.

Oznacza to, że do dziewięciu kopert powinniśmy włożyć kolejno: 1$, 2$, 4$, 8$..., 256 $. Do dziesiątej wkładamy resztę banknotów, tj. w sumie 489$ (=1000 – 511). Każdą liczbę nie większą niż 488 można zapisać jako sumę potęg liczby 2, co daje rozwiązanie (wykorzystamy co najwyżej dziewięć kopert).

Co z liczbami pomiędzy 489 a 1000? Można je zapisać jako sumę 489 i sumę potęg liczby 2, więc mamy rozwiązanie (wszystkie 10 kopert wykorzystamy tylko do zapisu liczby 1000).

c) W jaki sposób została zastosowana technika typu przekształć i zwyciężaj?

d) Dla jakich kwot pieniędzy rozwiązań łamigłówki jest więcej?

Zadanie 2 (Odważniki)

Zaprojektuj zestaw takich dziesięciu odważników do wagi szalkowej, który pozwoli na wyznaczenie ciężaru jak największej liczby ważonych przedmiotów z dokładnością do 1 kg.

Zakładamy, że odważniki można kłaść tylko na jednej szalce wagi.

Przykład 3 (Ciasto)

Jaka jest największa liczba kawałków, na jakie można podzielić prostokątne ciasto, przecinając je 11 razy? Każda z linii cięcia ma być równoległa do jednego z boków trójkąta.

a) Zaproponuj rozwiązanie „siłowe".

b) Jak przeformułować zadanie na zadanie optymalizacyjne dla funkcji kwadratowej?

Rozwiązanie:

Oznaczmy przez h liczbę cięć poziomych, a przez v liczbę cięć pionowych. Jakim wyrażeniem opiszemy całkowitą liczbę uzyskanych kawałków? To $(h + 1)(v + 1)$.

Ponieważ $h + v = 11$, więc pierwotny problem sprowadza się do wyznaczenia największej możliwej wartości iloczynu $(h + 1)(11 - h + 1) = h(11 - h) + 12$, gdzie $h < 12$.

Można w prosty sposób pokazać (jak?), że zadanie optymalizacyjne ma dwa rozwiązania (5, 6) oraz (6, 5). Dla pierwotnego problemu oznacza to, że rozwiązaniem jest liczba 30.

Zadanie 3 (Tramwaj)

Ania, Bartek, Darek, Hania, Krzyś, Robert, Sonia i Tomek dojeżdżają do szkoły tramwajami tej samej linii (niekoniecznie jadąc w tym samym kierunku). Wsiadają na przystankach: 4, 2, 4, 3, 12, 11, 11 i 10 (licząc dla tramwaju jadącego w tym samym kierunku). Okazuje się, że przystanek najbliższy szkole znajduje się w takim miejscu, że średnia długość ich podróży (mierzona przez liczbę przystanków) jest najmniejsza z możliwych. Który to przystanek?

a) Zaproponuj rozwiązanie „siłowe".

b) Jak przeformułować zadanie na zadanie dotyczącej statystyki?

Człowiek - najlepsza inwestycja

KAPITAŁ LUDZKI
NARODOWA STRATEGIA SPÓJNOŚCI

Warszawska
Wyższa Szkoła
Informatyki

UNIA EUROPEJSKA
EUROPEJSKI
FUNDUSZ SPOŁECZNY

Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego