

Metody projektowania algorytmów

Tworzenie projektów informatycznych opiera się w dużej mierze na formułowaniu i implementacji algorytmów, które mają za zadanie właściwe przetworzenie danych i rozwiązanie postawionych przed nami problemów. Algorytmy można sklasyfikować na kilka różnych sposobów, ale wśród nich najważniejszy jest podział ze względu na techniki ich konstruowania. Są pewne techniki tworzenia algorytmów, których zastosowanie prowadzi do efektywniejszego rozwiązywania problemów niż za pomocą algorytmów konstruowanych w sposób „spontaniczny”.

Metoda dziel i rządź

<https://pl.khanacademy.org/computing/computer-science/algorithms/merge-sort/a/divide-and-conquer-algorithms>

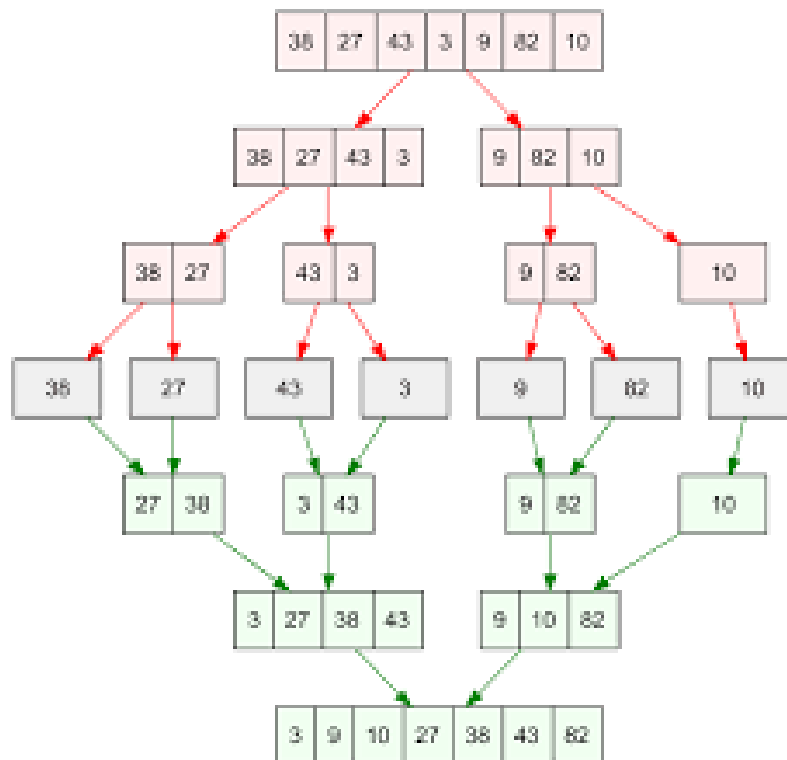
Dziel i rządź

- prowadzi nierzadko do bardzo efektywnych rozwiązań
- polega na rekurencyjnym dzieleniu problemu na dwa mniejsze podproblemy
- dzielenie ma miejsce tak długo aż podproblemy stają się proste do bezpośredniego rozwiązania
- zwykle podproblemy są „mniejszymi kopiami” podproblemu z którego powstały

Algorytm sprawdza czy w podanej, posortowanej tablicy znajduje się element o danej wartości.

- tablica jest dzielona na coraz mniejsze elementy (na pół)
- jako potencjalny element do wyszukania typuje się element środkowy
- w zależności od wartości elementu środkowego, kontynuuje się przeszukiwanie odpowiedniej części tablicy (zawężenie przedziału)
- podział kończy się gdy znajdziemy szukany element lub gdy przedział osiągnie długość 0

Przykład



Programowanie dynamiczne

- stosowane głównie do rozwiązywanie problemów optymalizacyjnych
- alternatywa dla pewnych zagadnień rozwiązywanych metodami zachłannymi

W odniesieniu do programowania opartego o „dziel i zwyciężaj”:

- Jeżeli podproblemy, na które został podzielony problem główny, nie są niezależne to w różnych podproblemach wykonywane są wiele razy te same obliczenia, warto jest wtedy zastosować ulepszenie tej metody jakim jest zastosowanie programowania dynamicznego

Zasady

- wyniki poszczególnych obliczeń są zapamiętywane w pomocniczej tablicy
- tablica ta jest wykorzystywana w kolejnych krokach
- eliminuje to konieczność wielokrotnego powtarzania tych samych obliczeń

Dla każdego podproblemu obliczenia są zatem wykonane tylko raz, a ich wynik jest zapamiętywany

Zastosowanie

- w automatach do kawy przy wydawaniu reszty w taki sposób, by monet było jak najmniej.
- algorytm Floyda-Warshalla (najkrótsze ścieżki między wszystkimi wierzchołkami w grafie)

Algorytmy zachłanne

Przykłady różnych alg.: https://eduinf.waw.pl/inf/utills/021_2021/1004.php

Problem wydawania reszty z opisem kodu: <http://kaj.uniwersytetradom.pl/cpp1.html>

- algorytm w każdym kroku dokonuje wyboru będącego na daną chwilę tym najlepszym (najbliższym końcowemu rozwiązaniu)
- podejmuje decyzje optymalne tylko lokalnie
- kontynuuje działania wynikające z poprzednich decyzji
- podejście często okazuje się nieoptymalne

Przykład

- Algorytm Kruskala (MST)
- Algorytm Dijkstry (najkrótsza ścieżka w grafie o nieujemnych wagach)