Pytania do egzaminu inżynierskiego z Informatyki – Rafał Bańka

Algorytmy i struktury danych

1. Podstawowe techniki algorytmiczne: metoda dziel i zwyciężaj, metoda zachłanna oraz metoda programowania dynamicznego.

   Metoda dziel i zwyciężaj – polega na dzielenie problemu na mniejsze pod-problemy, podobne do pierwszego. Następnie dane pod-problemy są dzielone na mniejsze części, aż do momentu, gdy staną się łatwe do rozwiązania. Na końcu pod-problemy są łączone, dając końcowy wynik.

   Metoda zachłanna – polega na podejmowaniu decyzji, które są najkorzystniejsze w danej chwili, licząc na znalezienie globalnego, optymalnego rozwiązania.

   Metoda programowania dynamicznego – podobnie jak w metodzie dziel i zwyciężaj, dany problem dzieli się na mniejsze pod-problemy. Jednakże metoda ta zapisuje wyniki rozwiązanych pod-problemów w tabeli. Dzięki czemu algorytm nie rozwiązuje ponownie tego samego pod-problemu. Metoda ta jest zwykle wykorzystywana w problemach optymalizacyjnych.

2. Wybrane algorytmy sortowania: przez wstawianie, wybór, bąbelkowe, mergesort, heapsort, quicksort; złożoność powyższych algorytmów.

   Sortowanie przez wstawianie – algorytm tworzy zbiór posortowany wyciągając dany element ze zbioru nieposortowanego. Następnie wyciągany jest kolejny nieposortowany element do zboru posortowanego, gdzie porównywany jest ze znajdującymi się w nim elementami. Jeśli wyciągnięty element nieposortowany jest mniejszy bądź równy elementowi posortowanemu, porównywanie zostaje zakończone i wyciągnięty element jest wstawiany przed tym posortowanym elementem. Złożoność czasowa algorytmu wynosi $O(n^2)$.

   Sortowanie przez wybór – w zbiorze elementów wybierany jest najmniejszy element, który zamieniany jest miejscami z pierwszym elementem tego zbioru. Wstawiony, najmniejszy element jest posortowany i pomija się go w dalszych obliczeniach.
   Złożoność czasowa algorytmu wynosi $O(n^2)$.

   Sortowanie bąbelkowe – algorytm rozpoczyna cykl i wybiera pierwszy element ze zbioru, a następnie porównuje go z sąsiednim elementem. Jeżeli pierwszy element jest większy od sąsiedniego, to elementy zostają zamienione miejscami. Cykl jest powtarzany do momentu gdy największy element jest na końcu listy zbioru. Algorytm rozpoczyna ponownie cykl i wykonuje tą samą czynność. Złożoność czasowa algorytmu wynosi $O(n^2)$.

   Sortowanie przez scalanie (Mergesort) – algorytm oparty jest na metodzie "Dziel i Zwyciężaj". Podstawowym pomysłem Mergesorta jest podział listy (lub tablicy) na dwie mniejsze części, rekurencyjne posortowanie każdej z tych części, a następnie scalenie ich w jedną, uporządkowaną listę. Proces ten jest powtarzany, aż lista zostanie w pełni posortowana. Proces ten kontynuowany jest aż do wyczerpania wszystkich elementów w obu podlistach. Złożoność czasowa wynosi $O(n \log n)$.

   Sortowanie szybkie - algorytm sortowania, który dzieli listę na części za pomocą wybranego elementu pivot, a następnie rekurencyjnie sortuje te części. Ma średnią złożoność czasową $O(n \log n)$, ale może osiągnąć $O(n^2)$ w najgorszym przypadku. Jest szybki i działa w miejscu, co oznacza, że nie wymaga dodatkowej pamięci.

3. Podstawowe algorytmy grafowe: najlżejsze ścieżki oraz przeszukiwania grafów.

   Najlżejsze ścieżki:
   Algorytm Dijkstry - znajduje najkrótsze ścieżki od jednego źródłowego wierzchołka do wszystkich pozostałych w grafie ważonym, gdzie krawędzie mają nieujemne wagi. Rozpoczyna od źródłowego wierzchołka i stopniowo odwiedza najbliższe wierzchołki, aktualizując odległości od źródła. Często stosowany w problemach związanych z trasowaniem w sieciach komunikacyjnych, planowaniem tras w logistyce itp.

   Algorytm Bellmana-Forda - znajduje najkrótsze ścieżki od jednego wierzchołka do wszystkich pozostałych w grafie ważonym, niekoniecznie z nieujemnymi wagami na krawędziach. Wykonuje relaksację na krawędziach wielokrotnie, by znaleźć optymalne ścieżki, aktualizując odległości między wierzchołkami. Używany w grafach z wagami o dowolnym znaku
   i w grafach z potencjalnie ujemnymi cyklami.

   Przeszukiwania grafów:
   Przeszukiwanie w szerz (Breadth-First Search – BFS) - przeszukuje graf w sposób wszerz, najpierw odwiedzając wszystkich sąsiadów danego wierzchołka, a następnie przechodząc do sąsiadów sąsiadów. Wykorzystuje kolejkę FIFO (First-In-First-Out) do odwiedzania wierzchołków w kolejności od najbliższych do najdalszych. Wykorzystywane do znajdowania najkrótszej ścieżki między dwoma wierzchołkami w nieskierowanym lub skierowanym grafie bez wag na krawędziach.

   Przeszukiwanie w głąb (Depth-First Search – DFS) - przeszukuje graf w sposób rekurencyjny, idąc jak głęboko się da wzdłuż jednej gałęzi zanim powróci do innych gałęzi. Wykorzystuje stos (stack) lub rekursję do odwiedzania wierzchołków. Wykorzystywane do znajdowania cykli w grafie, topologicznego sortowania, rozwiązywania problemów związanych z drzewami i grafami.

4. Algorytmy tekstowe: algorytm KMP oraz algorytm Karpa Rabina.

   Algorytm KMP - algorytm KMP sprawdza po kolei każdy znak tekstu, do którego możemy dopasować wzorzec. W momencie, gdy pierwszy znak wzorca oraz aktualnie przetwarzany znak tekstu są takie same, rozpoczynamy sprawdzanie po kolei, czy następne znaki wzorca i tekstu są identyczne. Na pierwszy rzut oka metoda ta działa w taki sam sposób, jak metoda naiwna. Usprawnienie pojawia się jednak w momencie, gdy już znaleźliśmy potencjalne dopasowanie, które okazuje się błędne. W takiej sytuacji algorytm KMP pomija sprawdzanie dopasowania na tych następnych pozycjach, które na pewno okażą się błędne.

   Algorytm Karpa Rabina - to technika do wyszukiwania wzorca w tekście, która wykorzystuje funkcję haszującą, aby przyspieszyć proces. Algorytm ten porównuje hasze fragmentów tekstu z haszem wzorca, co pozwala na efektywne znalezienie wszystkich wystąpień wzorca w tekście Jest używany w praktyce do znajdowania wzorców w tekście, a jego główne kroki to obliczanie haszy, porównywanie ich i przesuwanie okna po tekście. Algorytm jest używany w różnych zastosowaniach, takich jak przeszukiwanie plików lub analiza tekstu. Jednakże, warto zaznaczyć, że algorytm może wymagać dodatkowych środków ostrożności w przypadku kolizji haszy, gdzie różne fragmenty tekstu mają te same hasze.