Temat: Szukamy elementu najmniejszego i porządkujemy elementy.

Czytamy podręcznik str. 82 – 86

1. Zapoznaj się dodatkowo z informacjami dotyczącymi algorytmów.

2 PIERWSZY ALGORYTM – PRZESZUKIWANIE ZBIORU

Zaczniemy nasze rozważania od bardzo prostego problemu, który każdy z Was, jak i każdy człowiek, rozwiązuje wielokrotnie w ciągu dnia. Chodzi o znajdowanie w zbiorze elementu, który ma określoną własność. Oto przykładowe sytuacje problemowe.

Ćwiczenie 1. Opisz, na czym polega każdy z opisanych niżej problemów, wymieniając dane i poszukiwany wynik. Zaproponuj sposób znajdowania poszukiwanego elementu:

- znajdź najwyższego ucznia w swojej klasie; a jak zmieni się Twój algorytm, jeśli chciałbyś znaleźć w klasie najniższego ucznia?
- znajdź w swojej klasie ucznia, któremu droga do szkoły zabiera najwięcej czasu;
- znajdź najstarszego ucznia w swojej szkole; jak zmieni się Twój algorytm, gdybyś chciał znaleźć w szkole najmłodszego ucznia?
- znajdź największą kartę w potasowanej talii kart;
- znajdź najlepszego gracza w warcaby w swojej klasie (zakładamy, że wszyscy potrafią grać w warcaby);
- znajdź najlepszego tenisistę w swojej klasie.

Tego typu problemy pojawiają się bardzo często, również w obliczeniach komputerowych, i na ogół są rozwiązywane w dość naturalny sposób – przeglądany jest cały zbiór, by znaleźć poszukiwany element. Zastanowimy się, jak dobra jest to metoda, i czy może istnieje szybsza metoda znajdowania w zbiorze elementu o określonych własnościach.

Postawiony problem może wydać się zbyt prosty, by zajmować się nim na informatyce – każdy uczeń zapewne potrafi wskazać metodę rozwiązywania, polegającą na systematycznym przeszukaniu całego zbioru danych. Tak pojawia się metoda przeszukiwania ciągu, którą można nazwać przeszukiwaniem **liniowym**.

Założenia

Poczyńmy najpierw pewne założenia.

Założenie 1. Na początku wykluczamy, że przeszukiwane zbiory elementów są uporządkowane, np. klasa – od najwyższego do najniższego ucznia lub odwrotnie, szkoła – od najmłodszego do najstarszego ucznia lub odwrotnie, Gdyby tak było, to rozwiązanie problemów z ćwicz. 1 i im podobnych byłoby dziecinnie łatwe – wystarczyłoby wziąć element z początku albo z końca takiego uporządkowania.

Założenie 2. Przyjmujemy także, że nie interesują nas algorytmy rozwiązywania przedstawionych sytuacji problemowych, które w pierwszym kroku porządkują zbiór przeszukiwany, a następnie już prosto znajdują poszukiwane elementy – to założenie wynika z dalszych rozważań.

Z powyższych założeń wynika dość naturalny wniosek, że aby znaleźć w zbiorze poszukiwany element musimy przejrzeć wszystkie elementy zbioru, gdyż jakikolwiek pominięty element mógłby okazać się tym szukanym elementem.

Przy projektowaniu algorytmów istotne jest również określenie, jakie działania (operacje) mogą być wykonywane w algorytmie. W przypadku problemu poszukiwania szczególnego elementu w zbiorze wystarczy, jeśli będziemy umieli porównać elementy między sobą. Co więcej, w większości problemów w ćwicz. 1, porównanie elementów sprowadza się do porównania liczb, właściwych dla porównywanych elementów, a oznaczających: wzrost, czas na dojście do szkoły (np. liczony w minutach), wiek. Elementy zbiorów utożsamiamy więc z ich wartościami i wartości te nazywamy **danymi**, chociaż często prowadzimy rozważania w języku problemu, posługując się nazwami elementów: wzrost, wiek itp.

Rysunek 1. Wszechnica Poranna: Algorytmika i programowanie. Wprowadzenie do algorytmiki i programowania – wyszukiwanie i porządkowanie informacji Maciej M Sysło

2. **Dlaczego tworzymy specjalny algorytm wyszukiwania najmniejszego elementu w zbiorze?** Komputery, które operują ogromną ilością danych działają według pewnych algorytmów. Ani my, ani komputer parząc np. na poniższy zbiór liczb nie jesteśmy w stanie na pierwszy rzut oka stwierdzić, który z jego elementów jest najmniejszy lub największy. Najprostszym rozwiązaniem wydaje się sprawdzanie po kolei i przechowywanie w pamięci aktualnie najmniejszego znalezionego elementu.

2 8 15 99 56 23 48 25 15 69 8 7 1 5 64 694 15 3 6 94 2 6 9 4 315 185 6 1 5 6486 135 415 131 5 156 15 2 1 35 86 841 12 3 1 53 1 5 315 315 7 6 647 6 48 156 31 5 621 50 61 84 2 16 8 158 6 35 18 68 866 48 613 166 899 18 819 156 306 356 189 18 16 8 663 4 71 450 68 268 60 864 046 653 168 15 23 98 85 481 356 9 841 16 635 65 6 84 14 323 68 77 32 61 13 0 9165 156 15 76 31 61 36 894 1056 60 486 948 6125 81 6156 15 6 56 86 156 15 156 8 87 1513 68 1 86 1 353 8 61 156 86 86 145 315 196 1353 5186 179 65 916 3691 651 916 5

3. Algorytm wyszukiwania najmniejszego elementu w zbiorze.

7 5 12 3 100

Krok 1. $Minimum = 7$ (najmniejszą liczbą jest pierwszy element naszego zboru, innych jeszcze nie znamy)

Przychodzi liczba 5.

Krok 2. $Czy\ 5 < 7?\ TAK$ 😊 $Nowe\ minimum = 5$

Przychodzi liczba 12.

Krok 3. $Czy\ 12 < 5?\ NIE$ ☹ $Minimum\ nadal = 5$

Przychodzi liczba 3.

Krok 4. $Czy\ 3 < 5?\ TAK$ 😊 $Nowe\ minimum = 3$

Przychodzi liczba 100.

Krok 5. $Czy\ 100 < 3?\ NIE$ ☹ $Minimum\ nadal = 3$

Odpowiedź: Najmniejszym elementem naszego zbioru jest liczba 3.

4. Algorytm wyszukiwania największego elementu w zbiorze.

32 43 4 99 28 8

Krok 1. $Maksimum = 32$ (największą liczbą jest pierwszy element naszego zboru, innych jeszcze nie znamy)

Przychodzi liczba 43.

Krok 2. $Czy\ 43 > 32?\ TAK$ 😊 $Nowe\ maksimum = 43$

Przychodzi liczba 4.

Krok 3. $Czy\ 4 > 43?\ NIE$ ☹ $Maksimum\ nadal = 43$

Przychodzi liczba 99.

Krok 4. $Czy\ 99 > 43?\ TAK$ 😊 $Nowe\ maksimum = 99$

Przychodzi liczba 28.

Krok 5. $Czy\ 28 > 99?\ NIE$ ☹ $Maksimum\ nadal = 99$

Przychodzi liczba 8.

Krok 6. $Czy\ 8 > 99?\ NIE$ ☹ $Maksimum\ nadal = 99$

Odpowiedź: Największym elementem naszego zbioru jest liczba 99.

5. Porządkowanie zbioru metodą przez wybieranie (od najmniejszego do największego elementu).

Porządkowanie zbioru liczb od najmniejszej do największej **metodą przez wybieranie** polega na wyszukiwaniu w nim najmniejszej liczby i zamienieniu jej miejscami z liczbą, która jest aktualnie na początku zbioru, a następnie powtarzaniu tych czynności z pominięciem liczb już uporządkowanych.