

W kolejnych zadaniach zakładamy, że elementy listy można ze sobą porównywać (czyli np. są to liczby lub napisy).

1. Napisz program, który wyszukuje element minimalny w liście A. Użyj pętli while. Podaj złożoność swojego programu, mierzoną liczbą porównań.
2. Napisz program, który wyszukuje element minimalny w liście A. Użyj pętli while. Podaj złożoność swojego programu, mierzoną liczbą porównań.
3. Napisz program, który wyznacza najmniejszy indeks elementu minimalnego w liście A.
4. Napisz program, który wyznacza największy indeks elementu minimalnego w liście A.
5. Powtórz poprzednie 4 zadania, lecz dla elementu maksymalnego.
6. Podaj program, który jednocześnie wyznacza element minimalny i maksymalny w liście A. Nie stosuj żadnych wbudowanych funkcji do liczenia tych elementów. Podaj złożoność programu, mierzoną liczbą operacji arytmetycznych.
7. Załóżmy, że dany jest ciąg liczb całkowitych a_0, a_1, \dots, a_{n-1} , $n > 0$ w liście A. Napisz programy (funkcje dla każdego podpunktu) o złożoności liniowej, które jako argument mają tę listę i obliczają:
 - $\sum_{i=0}^{n-1} a_i$, czyli sumę wszystkich wyrazów tego ciągu,
 - $\prod_{i=0}^{n-1} a_i$, czyli iloczyn wszystkich wyrazów tego ciągu,
 - średnią arytmetyczną wszystkich wyrazów tego ciągu,
 - średnią arytmetyczną wszystkich dodatnich wyrazów tego ciągu,
 - $\sum_{k=0}^{n-1} \prod_{i=1}^k a_i$.

8. Napisz algorytm (z wykorzystaniem pętli while) wyszukiwania binarnego klucza key w posortowanej rosnąco liście A . Jeżeli klucz key znajduje się w liście A , to algorytm powinien zwrócić taki indeks k , że $A[k] = key$. Jeżeli klucz key nie znajduje się w liście A , to algorytm powinien zwrócić $None$. Jaka jest według Ciebie złożoność czasowa optymistyczna i pesymistyczna podanego algorytmu?
9. Załóżmy, że dane są dwie listy A i B liczb naturalnych posortowane rosnąco i liczba naturalna x . Podaj program, który sprawdza, czy istnieją takie $a \in A$ i $b \in B$ takie, że $x = a + b$. Postaraj się, aby program wykonywał jak najmniej operacji arytmetycznych.