Program

• Podstawowe algorytmy oraz biblioteka standardowa C++

• Elementy analizy algorytmów

• Podstawowe metody projektowania algorytmów

• Sortowanie

• Grafy -definicja i grafy szczególne (Eulera i Hamiltona)

• Algorytmy grafowe: DFS i BFS

• Znajdowanie najkrótszych dróg

• Znajdowanie najkrótszego drzewa rozpinającego

• Drzewa poszukiwań binarnych

• Drzewa czerwono-czarne

• Algorytmy tekstowe

• Algorytmy geometryczne na płaszczyźnie

• Algorytmy przeszukiwania z nawrotami i aproksymacyjne

Stos (LIFO)

Stos to liniowa struktura danych z operacjami odłożenia na stos (std:: push) oraz zdjęcia ze stosu (std:: top/pop).

Stos nazywany jest też buforem typu LIFO (ang. Last In –First Out).

Kolejka (FIFO)

Kolejka to liniowa struktura danych z operacjami ustawienia w kolejce (std:: push) oraz wyjścia z kolejki (std:: front/pop).

Kolejka nazywana jest też buforem typu FIFO (ang. First In –First Out).

Kolejka priorytetowa

Kolejka priorytetowa(gdy klucz = wartość) to struktura danych oferująca:

–wstawianie do kolejki elementu o zadanej wartości (std:: push)

–usunięcie elementu o ekstremalnej wartości (std:: pop)

–zwrócenie elementu o ekstremalnej wartości (std:: top)

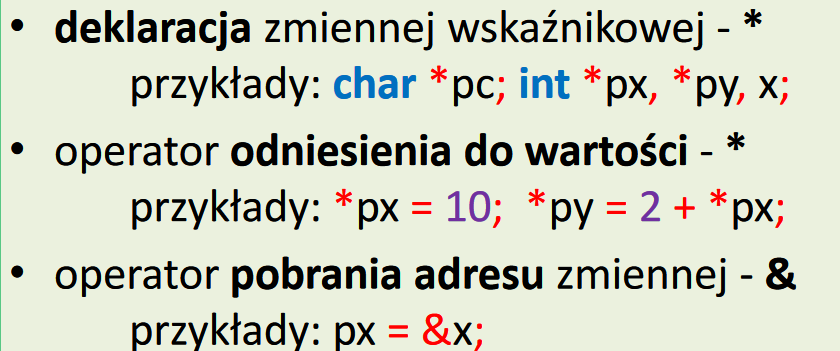
Kolejka priorytetowa jest wykorzystywana w wielu klasycznych algorytmach.

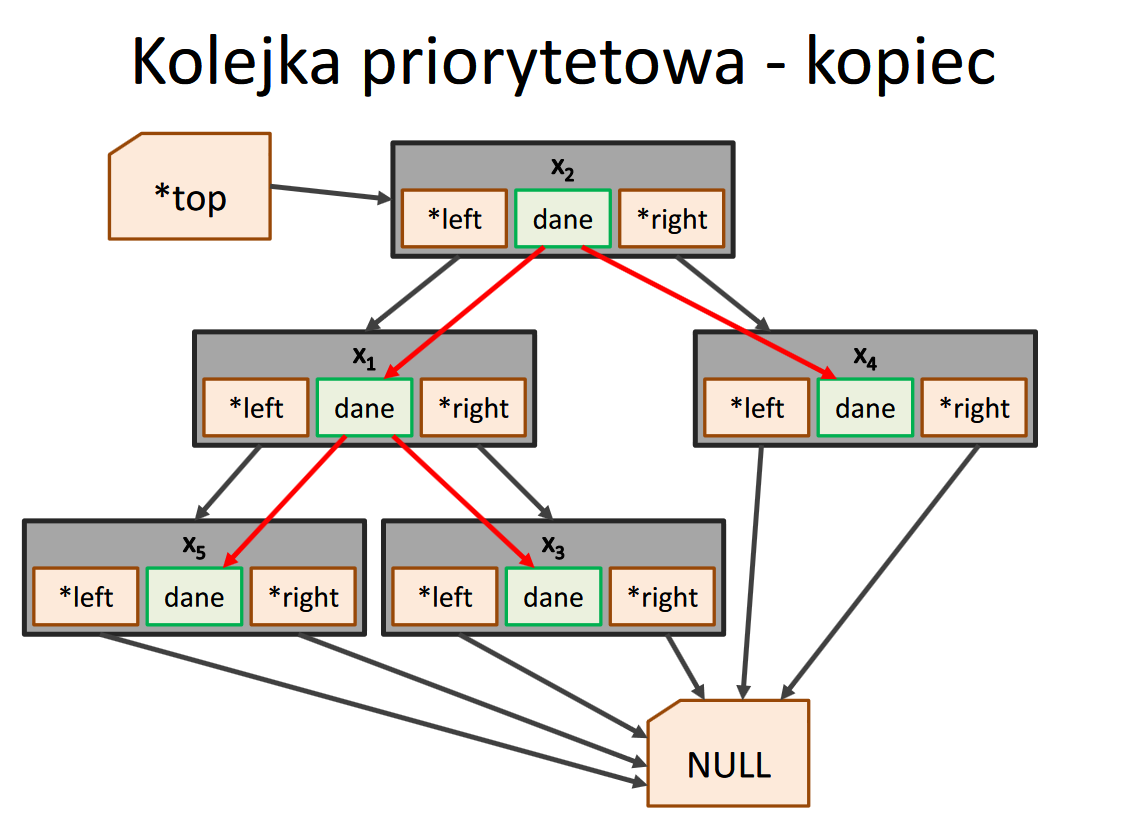
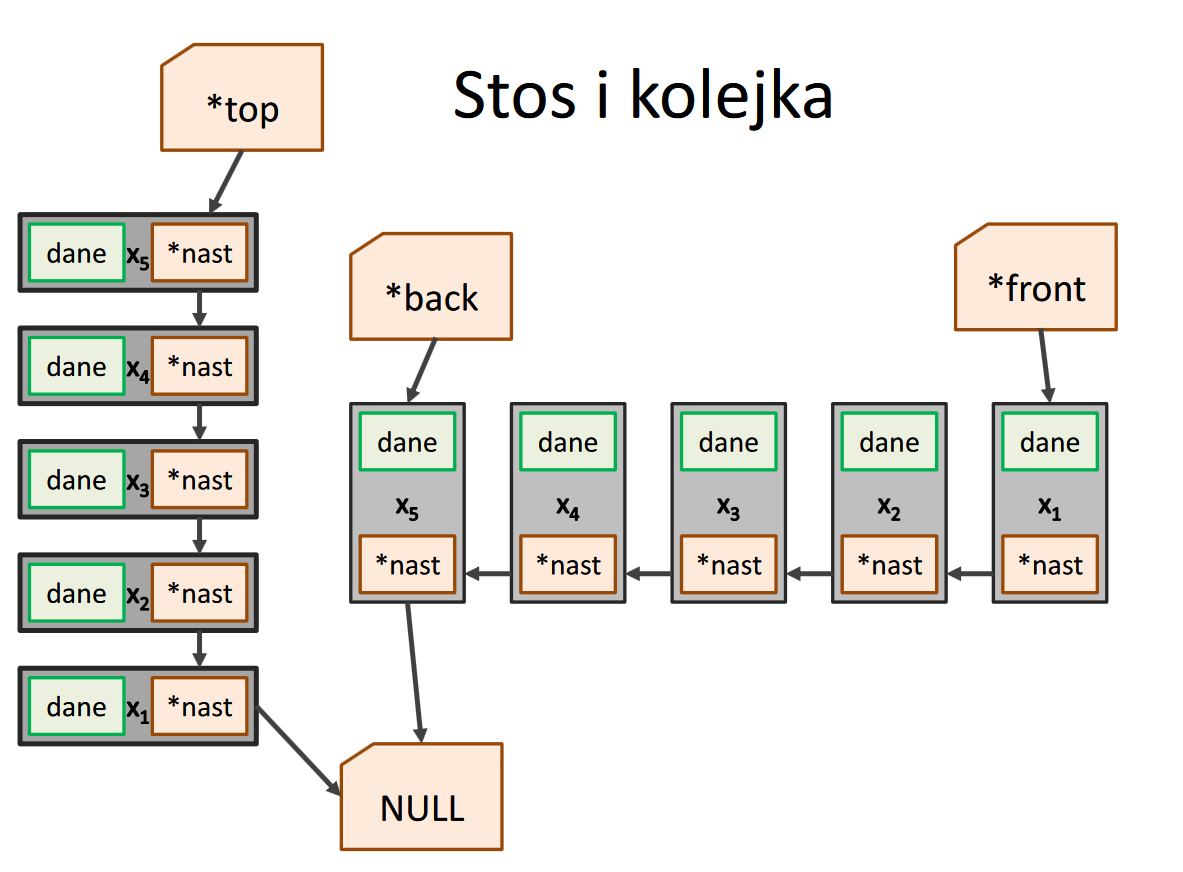
Wskaźniki -przypomnienie

•adres zmiennej to indeks pierwszego spośród bajtów w których zapisywana jest jej wartość

•wskaźnik to inaczej adres w pamięci

•zmienna wskaźnikowa to zmienna w której można przechowywać adresy



Biblioteka standardowa -STL

Właściwie standardowa biblioteka szablonów (standard template library)

Zawiera:

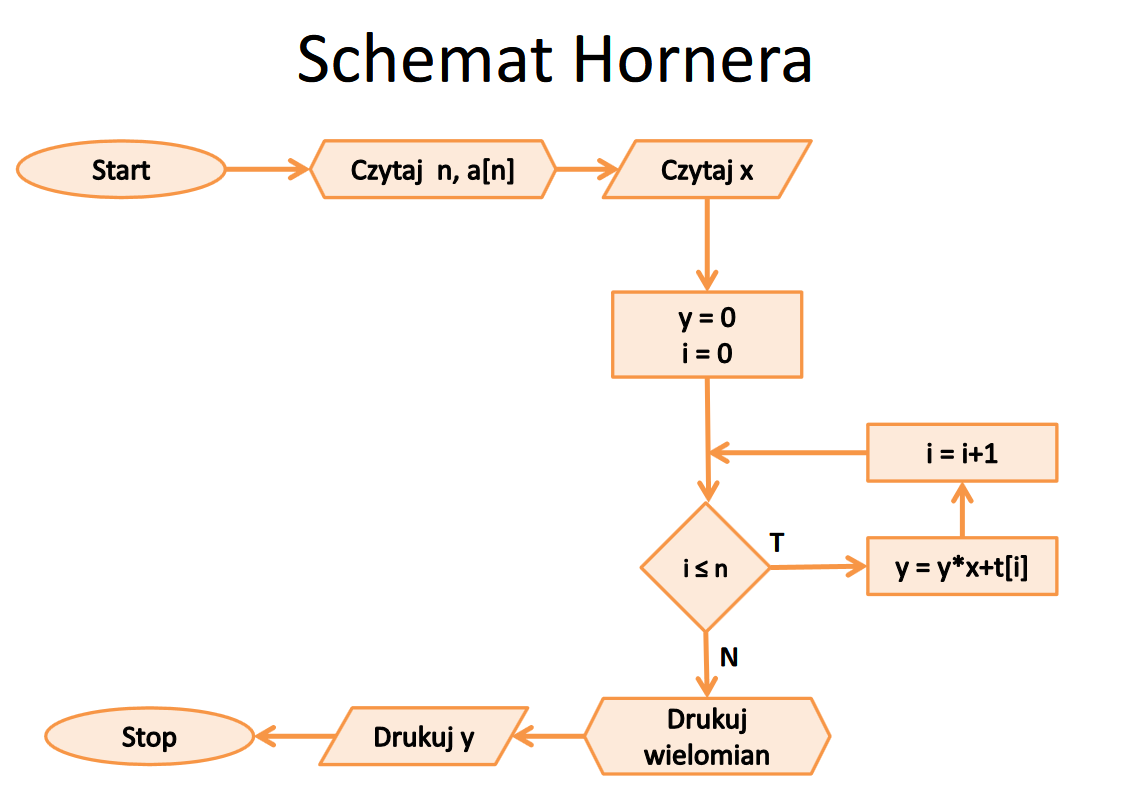
•kontenery(do przechowywania danych)

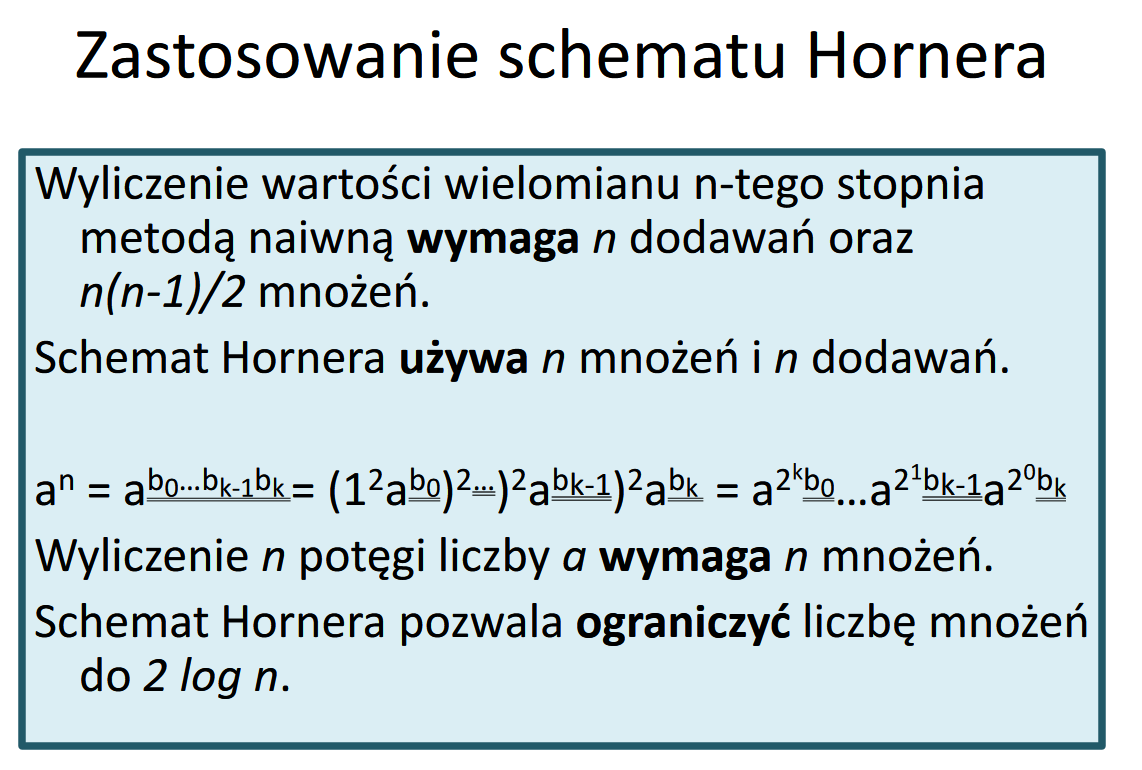
•adaptery(do zmiany interfejsu kontenerów)

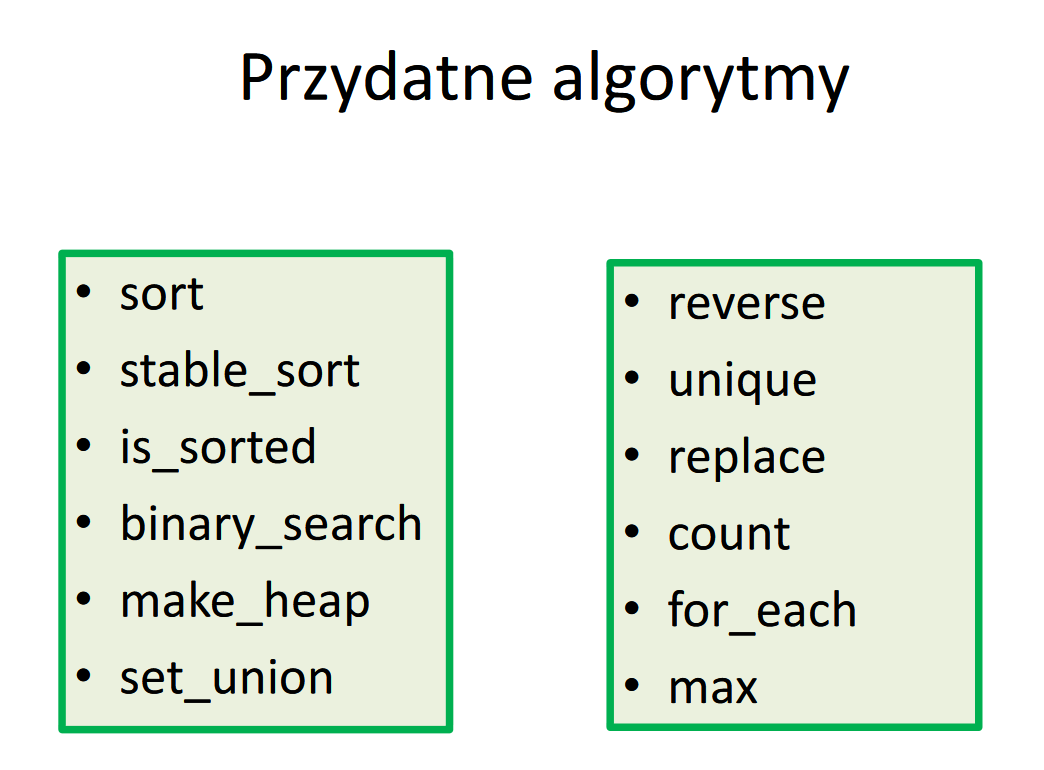
•iteratory (do poruszania się wewnątrz kontenerów)

•algorytmy(do prostych obliczeń na kontenerach)

•funktory(opakowujące funkcje w obiekt)







Algorytm -przypomnienie

Algorytm to skończony ciąg precyzyjnie zdefiniowanych czynności prowadzących do rozwiązania problemu, osiągnięcia zamierzonego celu

Definicja problemu (specyfikacja)

•dane i wynik (typ): co otrzymamy i czego się spodziewamy

•rozmiar danych: złożoność –czasowa i pamięciowa

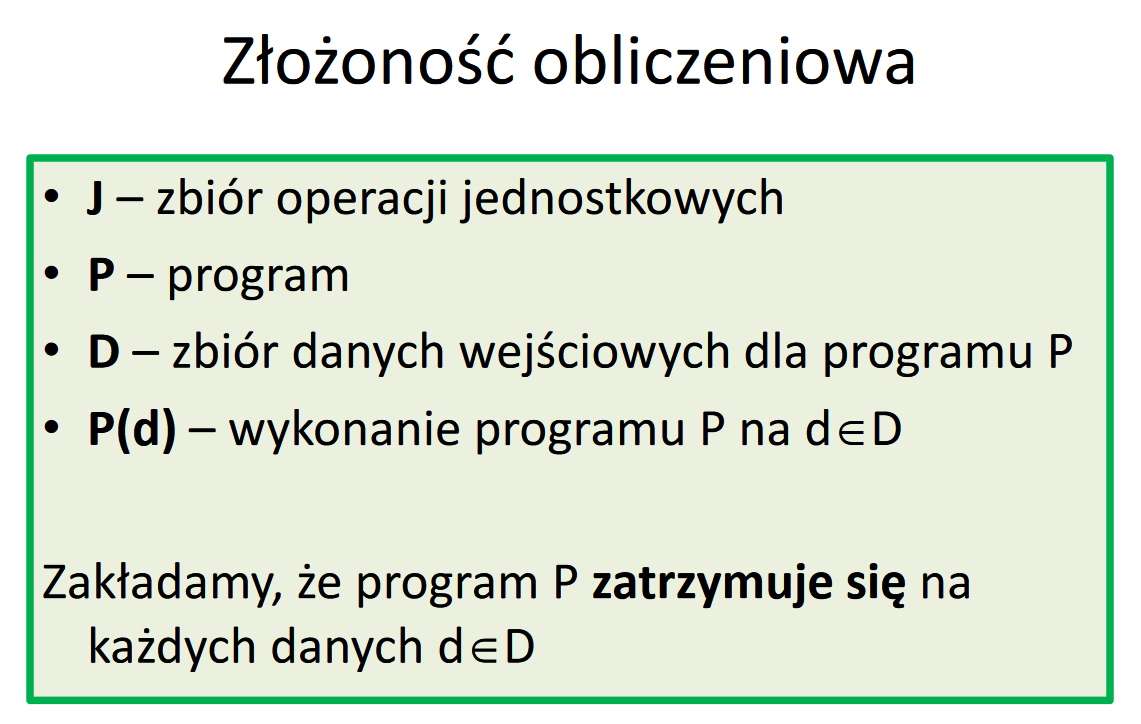
•definicja formalna: poprawność

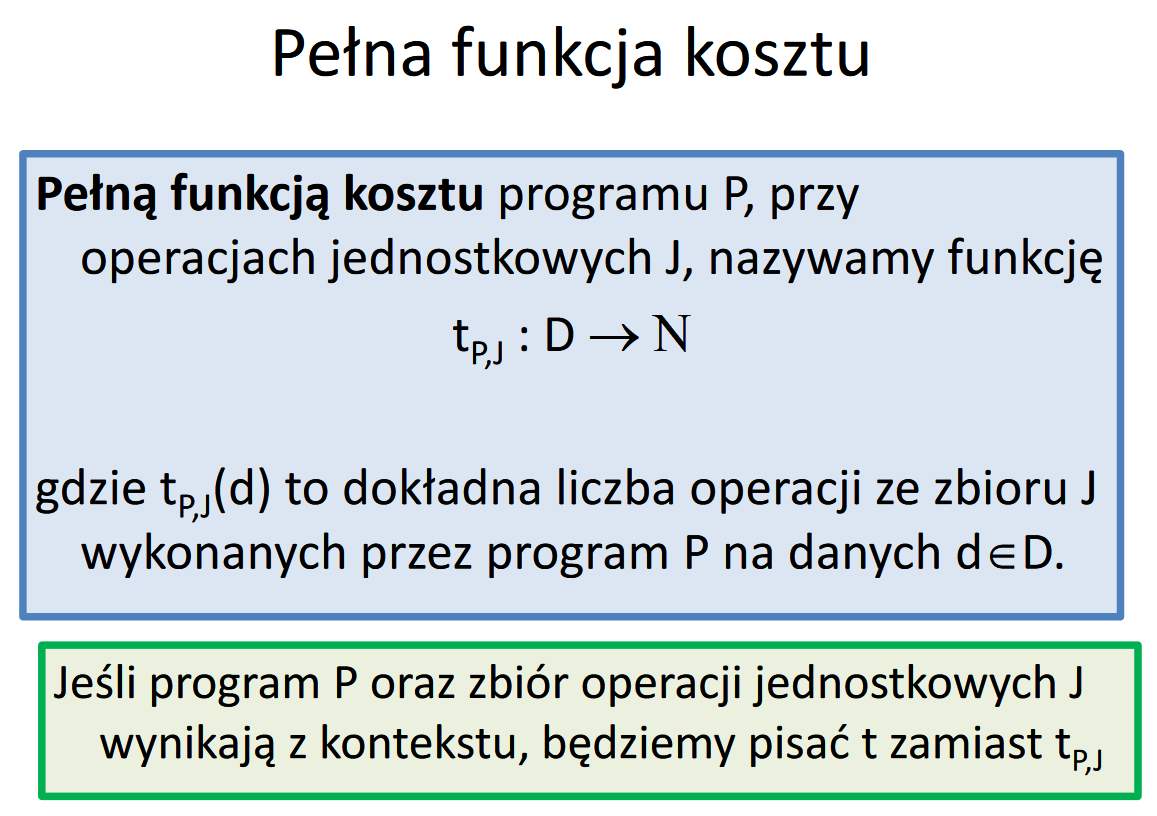
Złożoność obliczeniowa

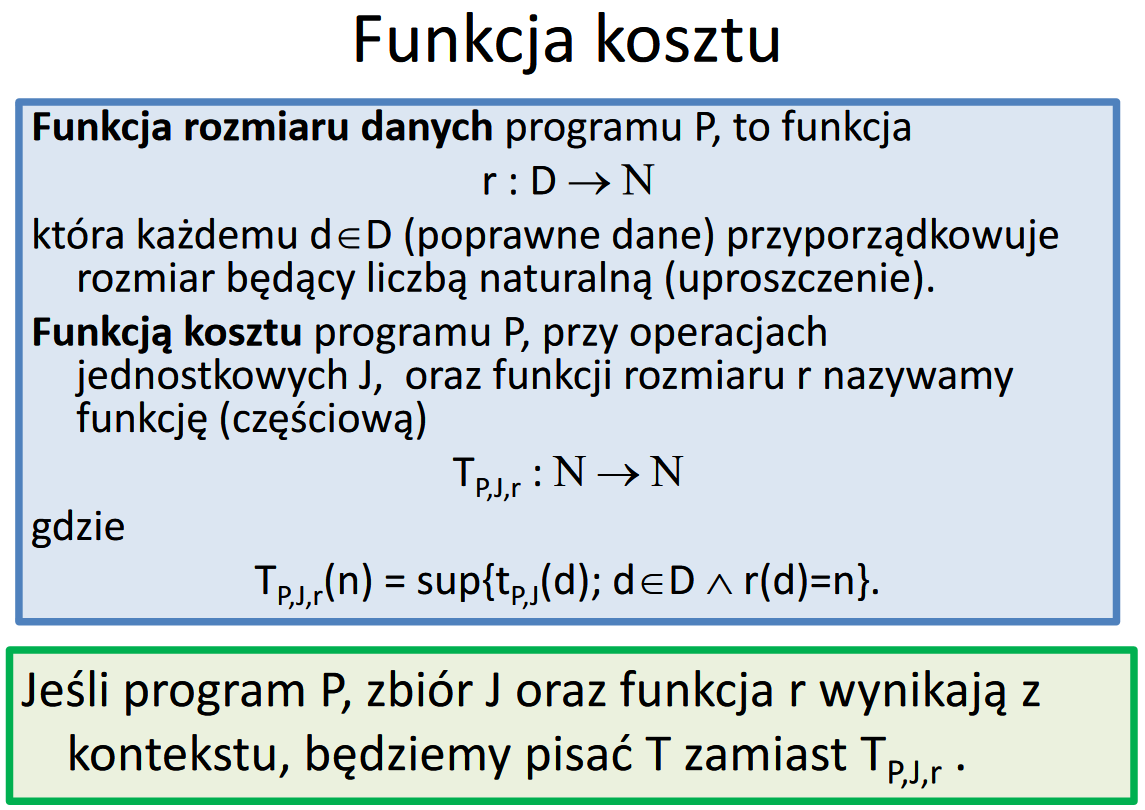
Złożoność czasowa algorytmu to liczba operacji jaka jest wykonywana na danych wejściowych wyrażona jako funkcja zależna od rozmiaru tych danych.

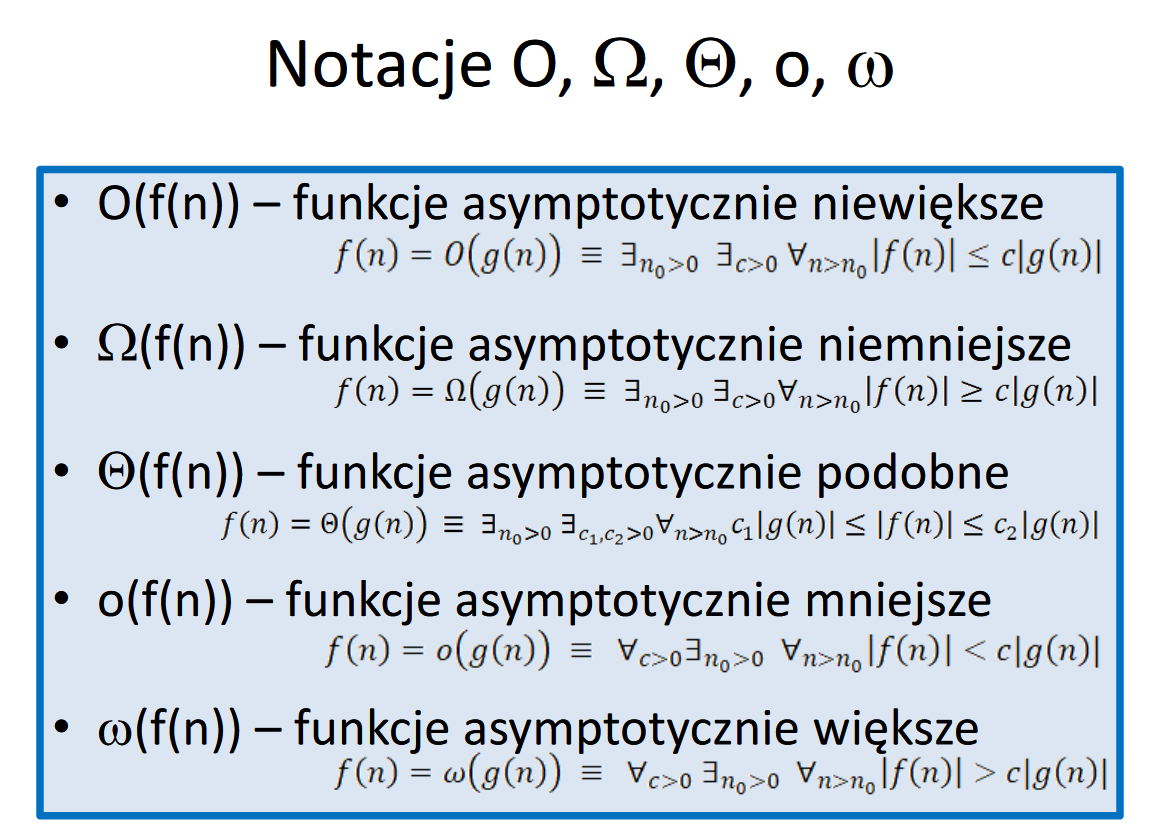
Jakie operacje brać pod uwagę?

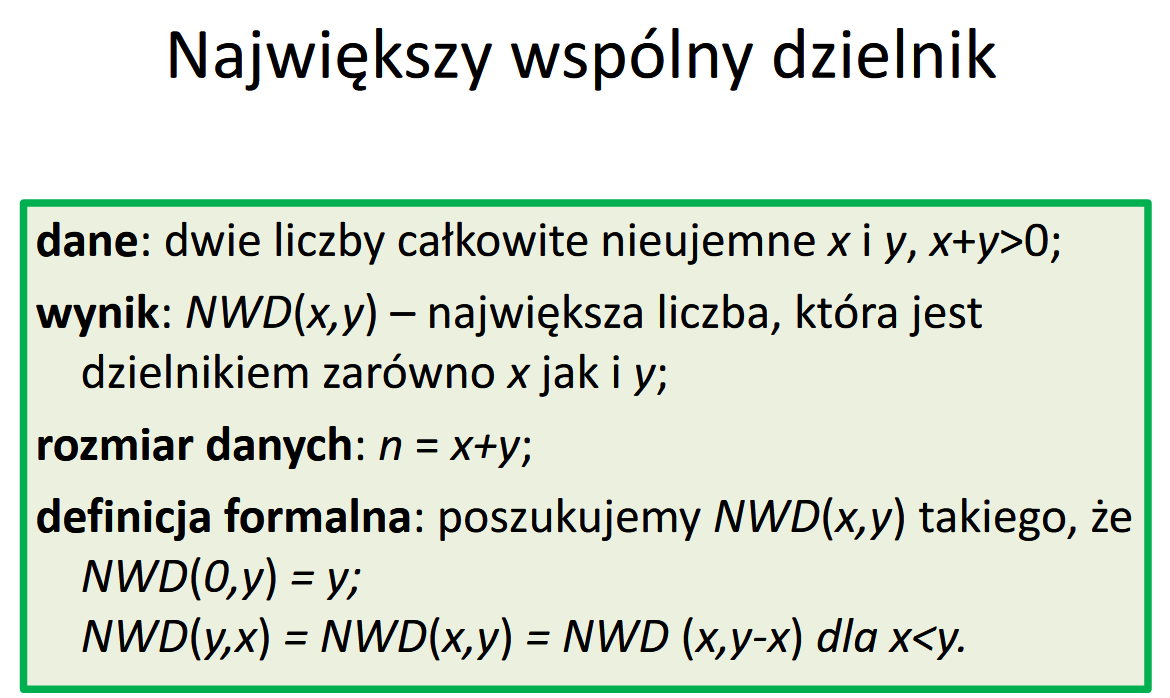
J - zbiór operacji jednostkowych, czyli operacji arytmetycznych, logicznych, instrukcji przypisania etc.











Koszt zamortyzowany

„Fundusz Ubezpieczeń Kosztów Algorytmicznych”

•Rozpoczynamy z bilansem 0

•Każda operacja wpłaca stałą kwotę f(n)

•Każda operacja pobiera tyle jednostek, ile wynosi jej faktyczny koszt

•Jeśli dla żadnych danych FUKA nie zbankrutuje, to operacja ma czas zamortyzowany f(n)

Dla uproszczenia, liczymy tylko operacje stosowe, top + pop jako jedna operacja!

Koszt zamortyzowany

•Rozpoczęliśmy z funduszem równym 0

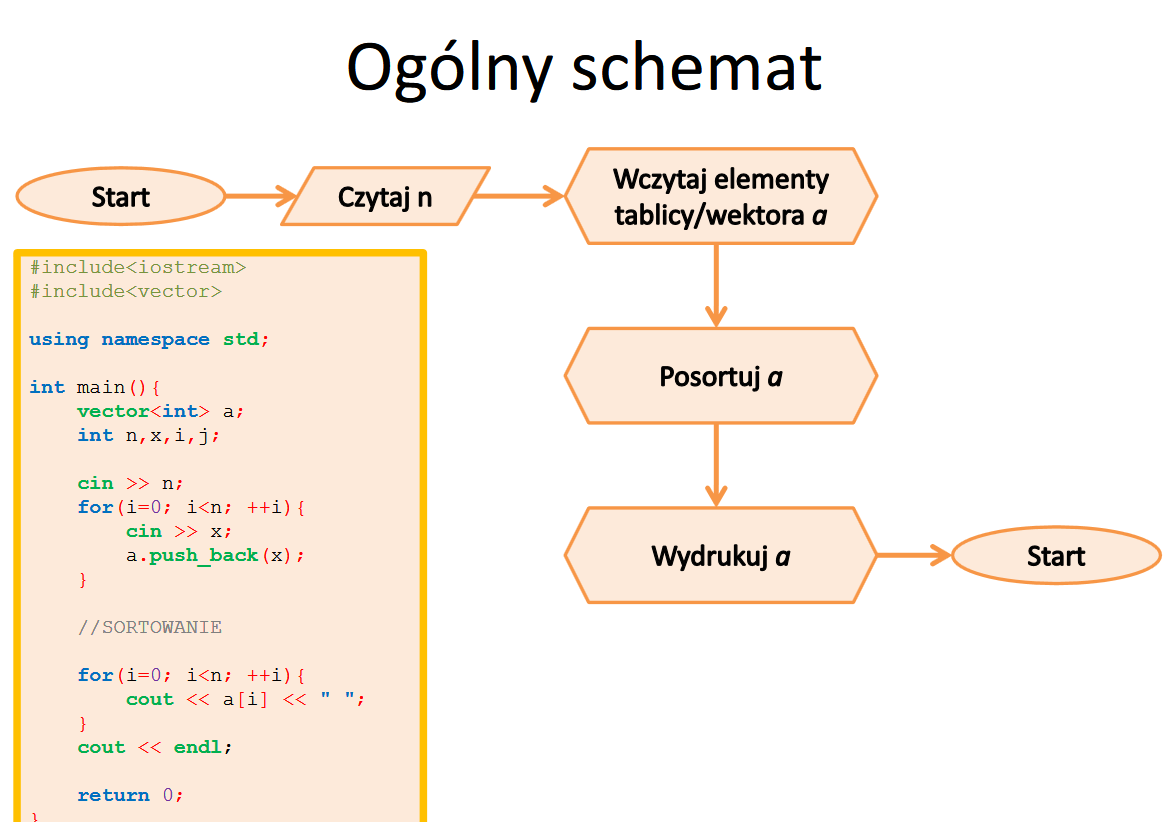
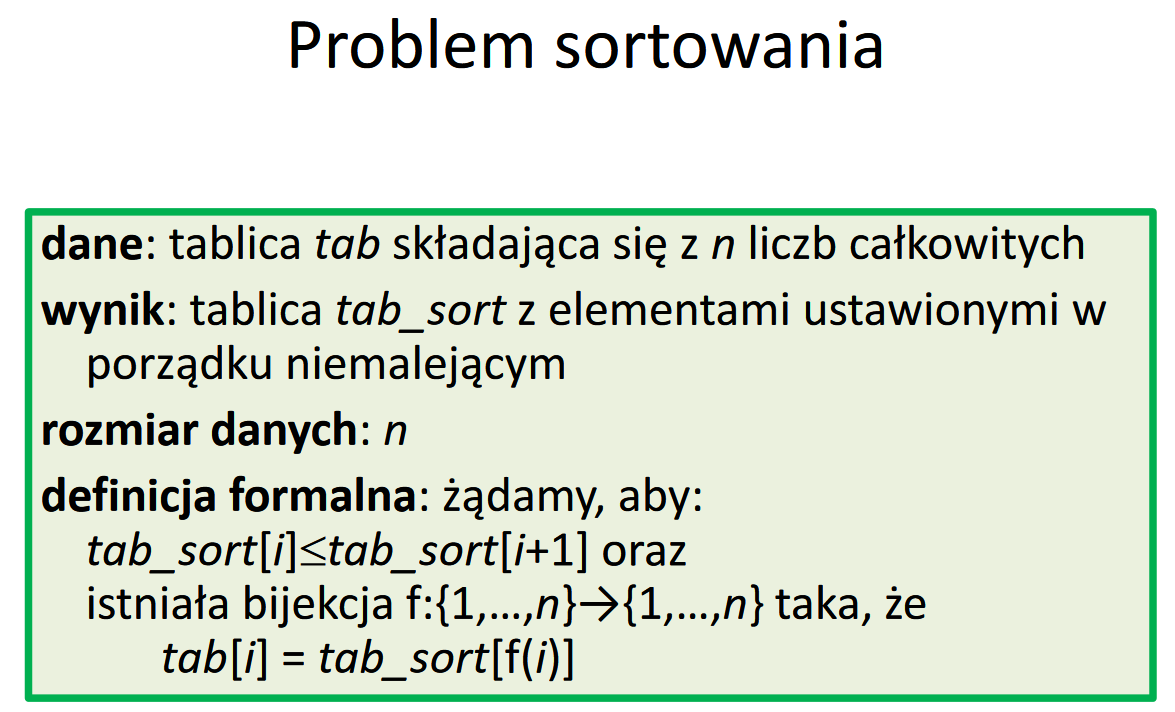
•Każde dodanie nowego elementu to wniesienie stałego kapitału (3= 4 -1)

•Każde przeniesienie obniża kapitał o 2

•Każde usunięcie obniża kapitał o 1

•Całkowita wartość kapitału to 3\*liczba elementów na pierwszym stosie+ liczba elementów na drugim stosie

•Zamortyzowany koszt dodania/usunięcia jest stały



Sortowania kwadratoweProste sortowania:

•Sortowanie bąbelkowe

•Sortowanie przez wybieranie

•Sortowanie przez wstawianie

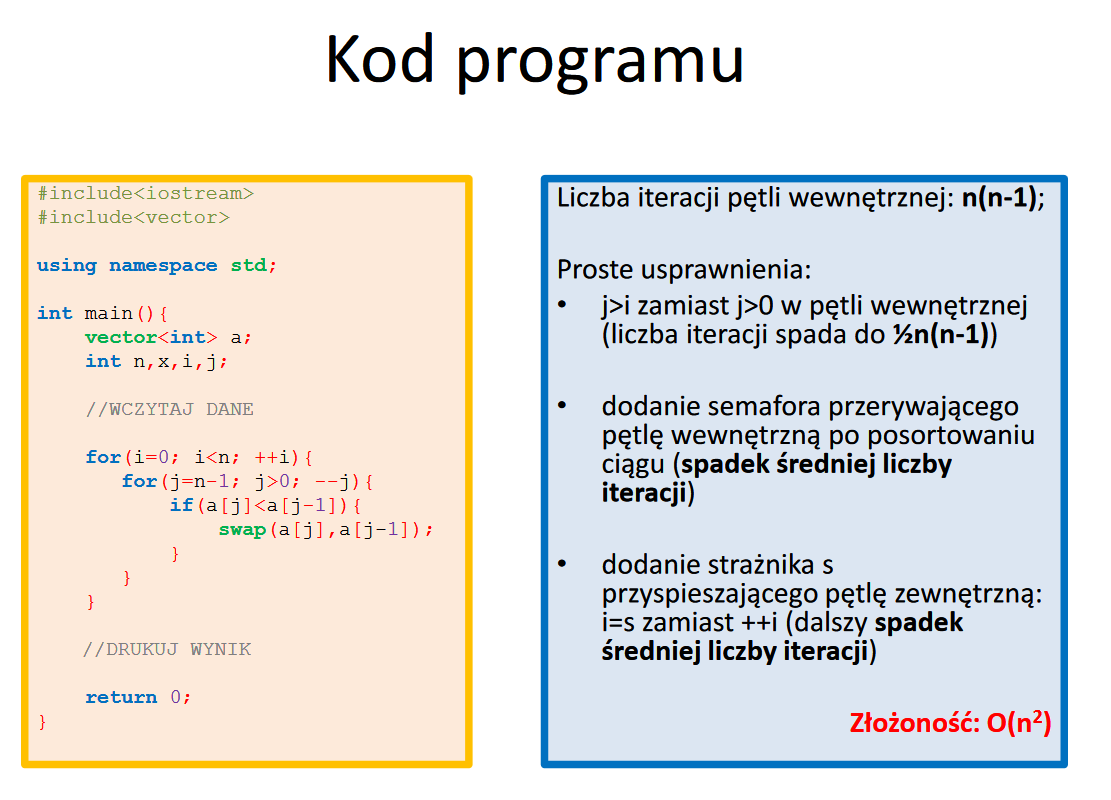
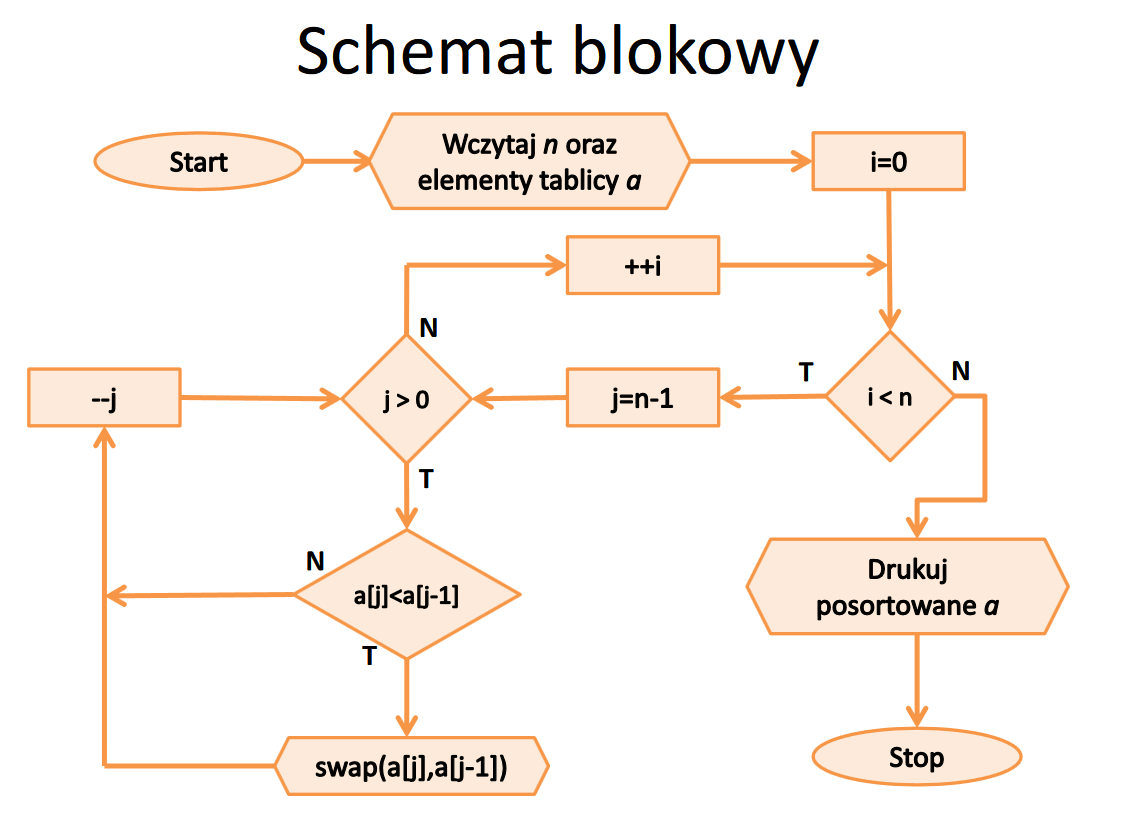
Wszystkie algorytmy, w kolejnych krokach, mają uporządkowany prefiks sortowanej tablicy.

Sortowanie bąbelkowe

Idea sortowania:

Porównuj kolejne pary sąsiadujących elementów poprawiając ich ustawienie, jeśli tworzą inwersję (występują w złym porządku).

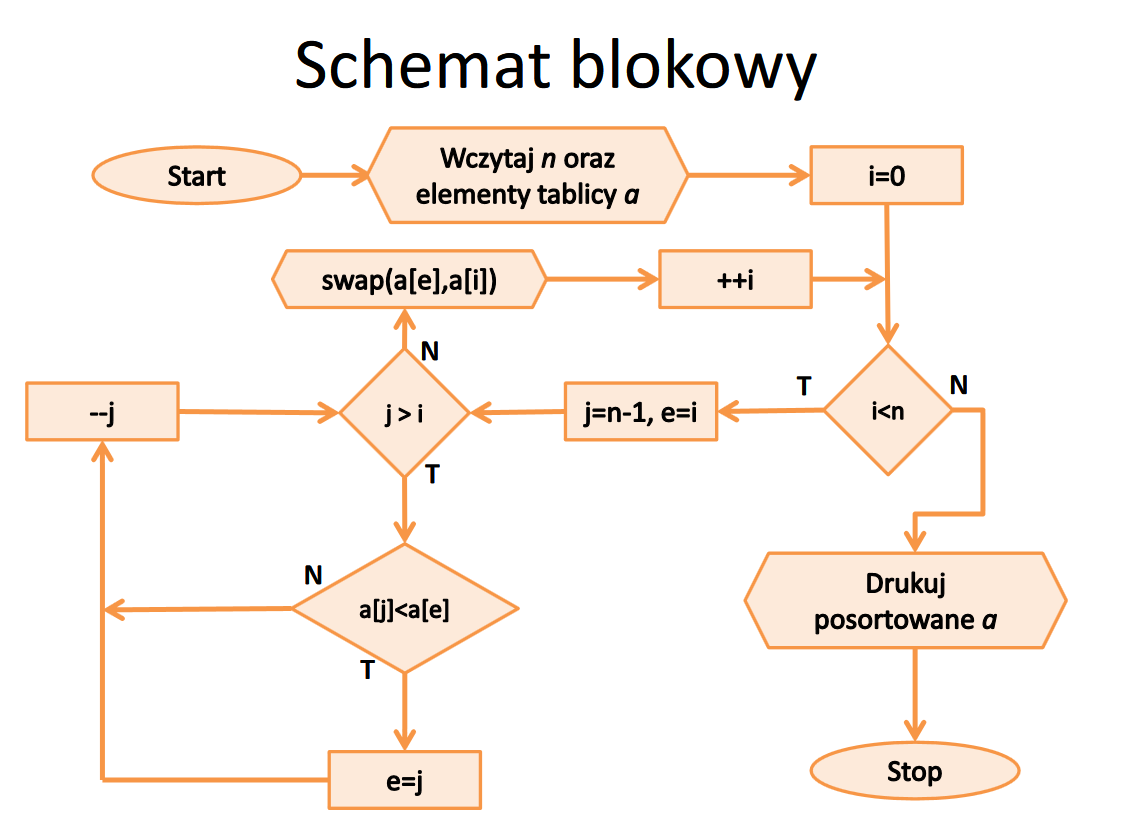
W najprostszym podejściu, można n-1 razy powtórzyć porównanie wszystkich par sąsiadujących elementów.

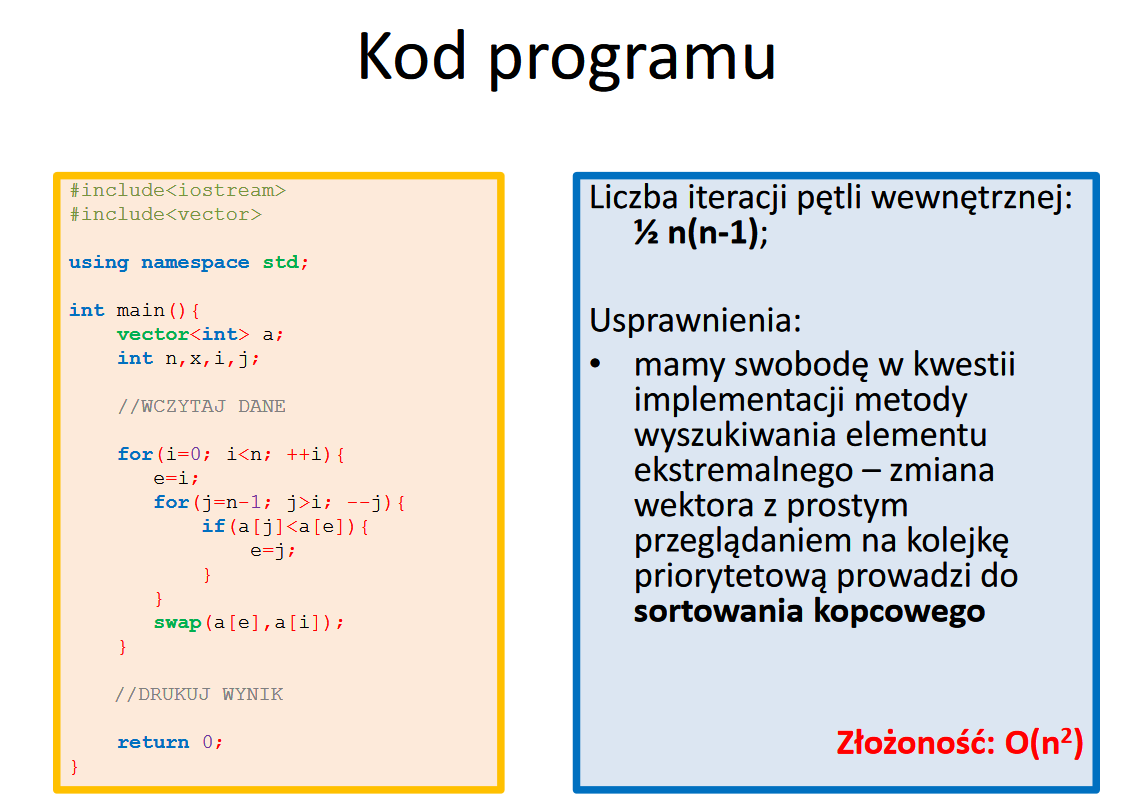


Sortowanie przez wybieranie

Idea sortowania: W nieposortowanej części wyszukaj element ekstremalny i zamień go z pierwszym elementem w tej części.

Algorytm znany Podstaw Programowania, z Wykładu 4. Wyszukania w nieposortowanym zbiorze jest liniowo zależne od jego rozmiaru.

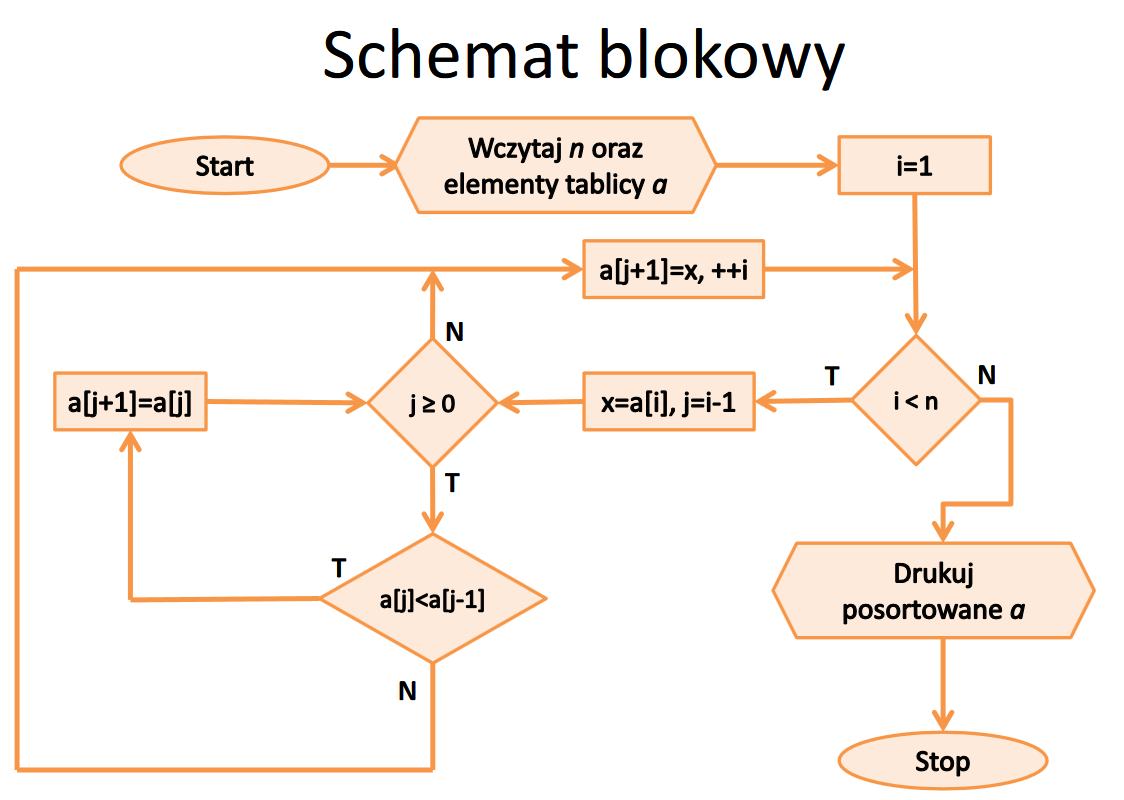


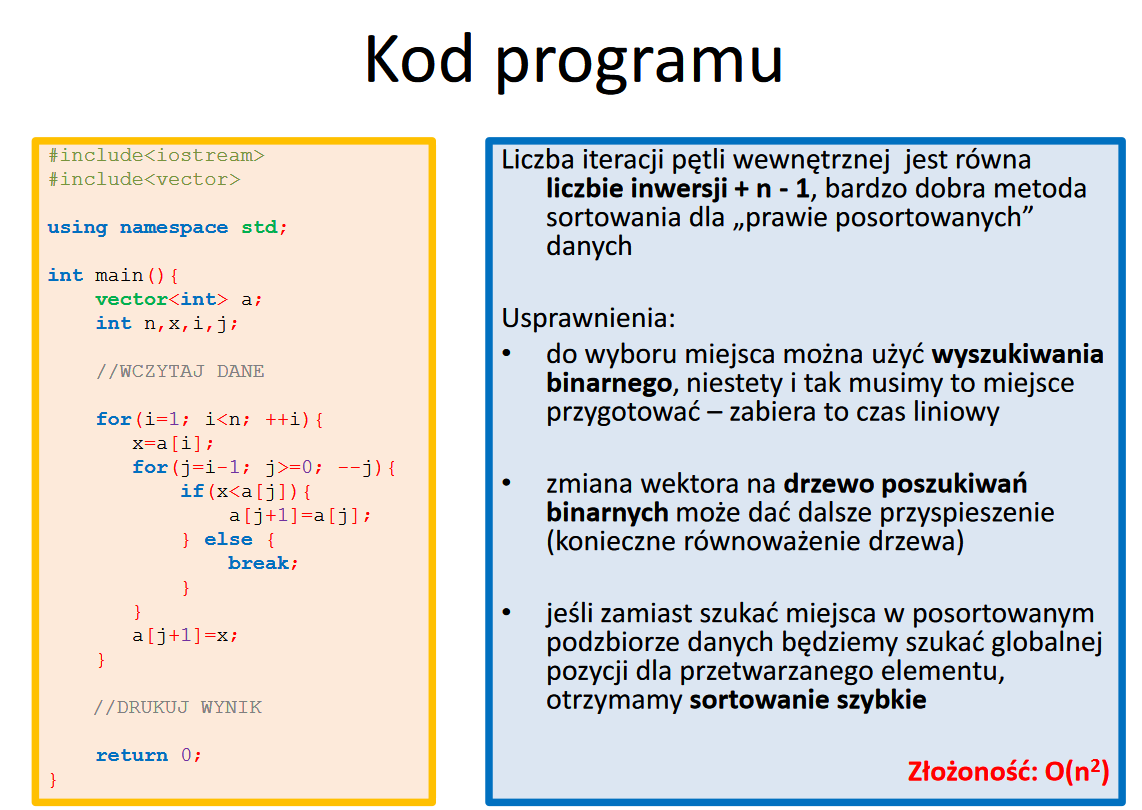


Sortowanie przez wstawianie

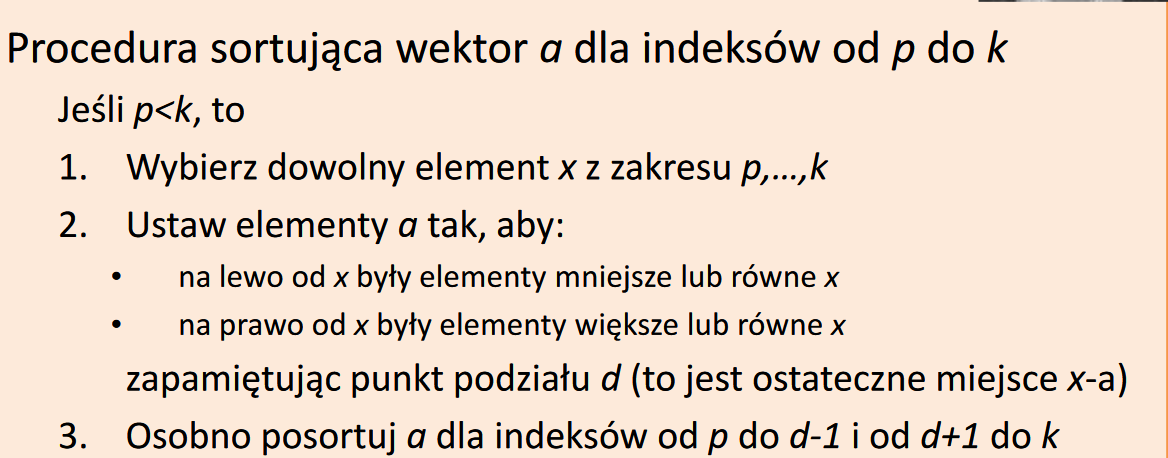
Idea sortowania: Weź dowolny element z nieposortowanej części i wyszukaj dla niego miejsce w części posortowanej, ustaw go w tym miejscu.

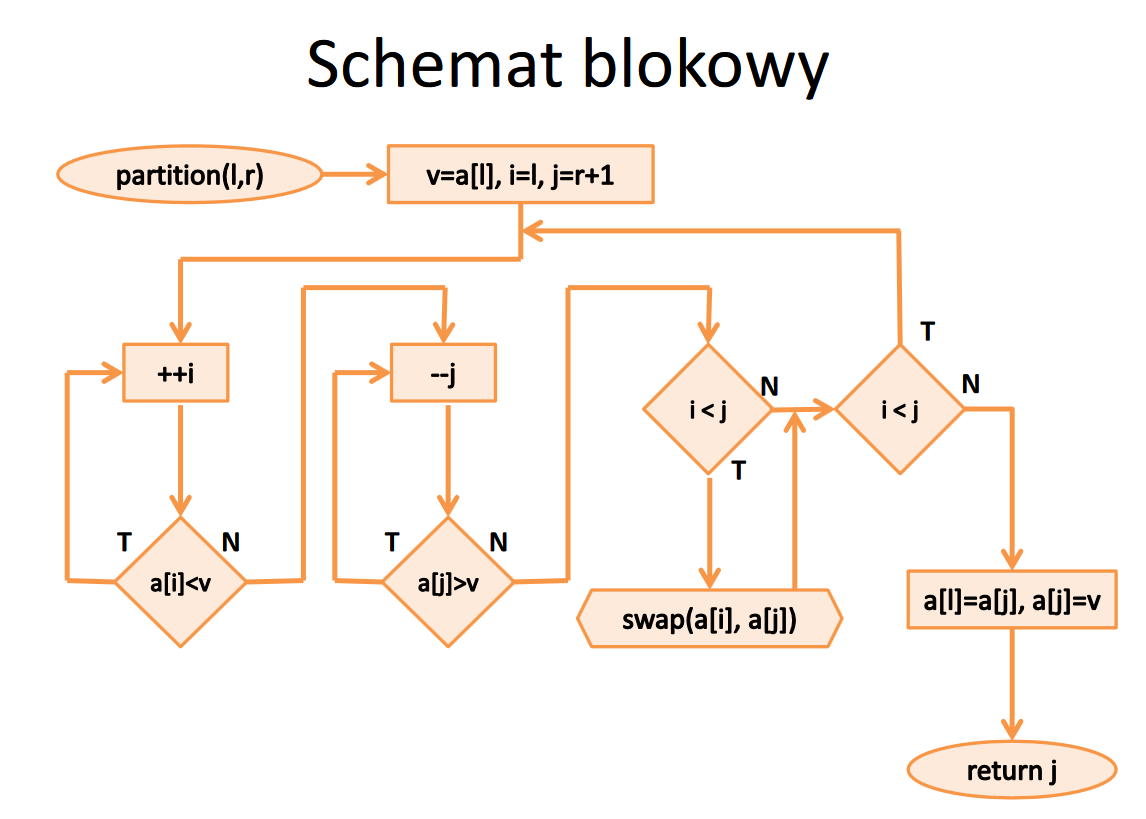
Podobnie jak w poprzednich przypadkach zakładamy, że po i-tej fazie a[0...i-1] jest posortowane.





Sortowanie szybkie





Sortowanie szybkie

•W przypadku złośliwych danych złożoność wynosi O(n2)

•Dane losowe sortowane są w czasie O(n log n)

•Sortowanie wymaga dodatkowego stosu, rozmiar stosu może być liniowy

Usprawnienia:

•Zastosowanie med3 zmniejsza prawdopodobieństwo wystąpienia złośliwych danych

•Jeśli na stos odkładać będziemy zawsze parę indeksów, które leżą dalej, jego rozmiar będzie logarytmiczny

Sortowania liniowo-logarytmiczne

•Sortowanie przez scalanie dwie wersje: Wykład 12 (PP)Wykład 2 (AiSD)

•Sortowanie z użyciem kolejki priorytetowej Wykład 14 (PP) -kopiec

Złożoność problemu sortowania

•Jest dokładnie n! różnych permutacji zbioru n- elementowego

•W drzewie decyzyjnym jest co najmniej n! liści, jest to drzewo binarne, czyli jego wysokość to co najmniej log2(n!)

•log(n!) = O(n log n), czyli algorytm sortujący oparty o drzewo decyzyjne musi wykonać O(n log n) porównań

Sortowanie w czasie liniowym

•Sortowanie kubełkowe

•Sortowanie przez zliczanie

•Sortowanie pozycyjne

Sortowania o złożoności liniowej korzystają ze struktury sortowanego zbioru

Sortowanie kubełkowe

Idea sortowania kubełkowego:

Mając n różnych kluczy po których sortujemy, możemy:

–stworzyć n list

–przejrzeć cały zbiór do posortowania

– elementy z kluczem k wrzucać do listy o numerze k

–złączyć listy

Takie sortowanie jest stabilne(elementy o równych kluczach nie zmieniają położenia względem siebie).

Sortowanie przez zliczanie

Idea sortowania przez zliczanie:

Mając nróżnych kluczy po których sortujemy, możemy:

–stworzyć tablicę/wektor rozmiaru n

–przejrzeć cały zbiór do posortowania licząc ile jest wystąpień każdego z kluczy

–w drugim przebiegu przepisać elementy do nowego kontenera od razu w dobre miejsca

Takie sortowanie jest stabilne.

Sortowanie pozycyjne

Idea sortowania pozycyjnego:

Mając dane w których elementy są ciągami długości k obiektów pochodzących ze zbioru n-elementowego, możemy:

–stworzyć tablicę/wektor rozmiaru n

–k razy wykonać stabilne sortowanie przez zliczanie na kolejnych pozycjach, począwszy od najmniej istotnej

Takie sortowanie można wykorzystać dla liczb k-cyfrowych, napisów, kodów pocztowych etc.

Sortowanie pozycyjne dużego zbioru liczb

Mamy do posortowania n liczb zapisanych binarnie o nie więcej niż log2nbitachkażda.Ile czasu zajmie posortowanie tego zbioru za pomocą sortowania pozycyjnego?

Każda faza sortowania pozycyjnego jest liniowa, jest log2n faz –po jednej dla każdego bitu, czyli złożoność tego rozwiązania to O(n log2n)!!!

STL -algorytm sort

Czy można wykorzystać sortowanie zaimplementowane w bibliotece standardowej do niestandardowego sortowania?

Spróbujmy posortować pary liczb po pierwszej współrzędnej (jeśli pierwsze współrzędne są identyczne, to weźmiemy pod uwagę drugą).