

Algorytmy Grafowe

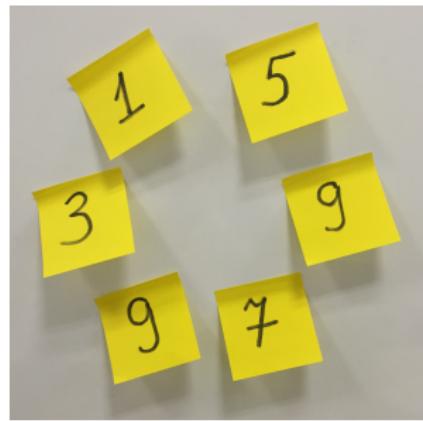
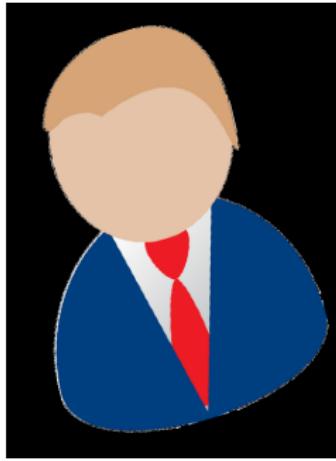
dr hab. Bożena Woźna-Szcześniak, prof. UJD

Uniwersytet Jana Długosza w Częstochowie

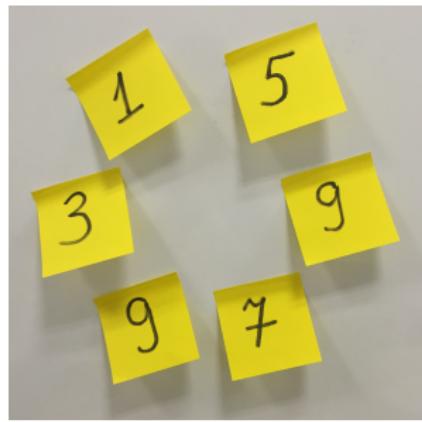
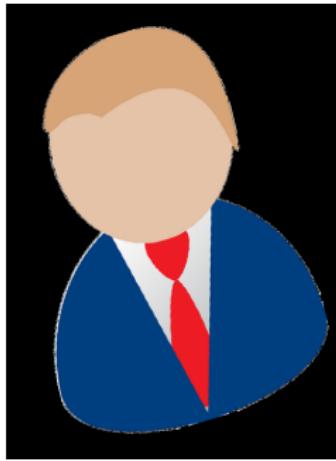
b.wozna@ujd.edu.pl

Wykład 9

Rozmowa kwalifikacyjna



Rozmowa kwalifikacyjna



Największa liczba

Definicja problemu

Jaka jest największa liczba, która składa się z cyfr 3, 9, 5, 9, 7, 1?
Użyj wszystkich cyfr.

Największa liczba

Definicja problemu

Jaka jest największa liczba, która składa się z cyfr 3, 9, 5, 9, 7, 1?
Użyj wszystkich cyfr.

Przykład

359179, 537991, 913579, ...

Największa liczba

Definicja problemu

Jaka jest największa liczba, która składa się z cyfr 3, 9, 5, 9, 7, 1?
Użyj wszystkich cyfr.

Przykład

359179, 537991, 913579, ...

Poprawna odpowiedź

997531 ...

Strategia zachłanna

- Dany jest zbiór cyfr:

5	7	3	9	1	9
---	---	---	---	---	---

Strategia zachłanna

- Dany jest zbiór cyfr:

5	7	3	9	1	9
---	---	---	---	---	---

- Znajdź maksymalną cyfrę

5	7	3	9	1	9
---	---	---	---	---	---

Strategia zachłanna

- Dany jest zbiór cyfr:

5	7	3	9	1	9
---	---	---	---	---	---

- Znajdź maksymalną cyfrę

5	7	3	9	1	9
---	---	---	---	---	---

- Dołącz ją do poszukiwanej liczby

5	7	3	9	1	9	→	9
---	---	---	---	---	---	---	---

Strategia zachłanna

- Dany jest zbiór cyfr:

5	7	3	9	1	9
---	---	---	---	---	---

- Znajdź maksymalną cyfrę

5	7	3	9	1	9
---	---	---	---	---	---

- Dołącz ją do poszukiwanej liczby

5	7	3	9	1	9	→	9
---	---	---	---	---	---	---	---

- Usuń ją z listy cyfr

5	7	3	1	9	→ 9
---	---	---	---	---	-----

Strategia zachłanna

- Dany jest zbiór cyfr:

5	7	3	9	1	9
---	---	---	---	---	---

- Znajdź maksymalną cyfrę

5	7	3	9	1	9
---	---	---	---	---	---

- Dołącz ją do poszukiwanej liczby

5	7	3	9	1	9	→	9
---	---	---	---	---	---	---	---

- Usuń ją z listy cyfr

5	7	3	1	9	→ 9
---	---	---	---	---	-----

- Powtarzaj, dopóki na liście znajdują się cyfry

Strategia zachłanna

- Znajdź maksymalną cyfrę:

5	7	3	1	9	→ 9
---	---	---	---	---	-----

Strategia zachłanna

- Znajdź maksymalną cyfrę:

5	7	3	1	9	→ 9
---	---	---	---	---	-----
- Dołącz ją do poszukiwanej liczby:

5	7	3	1	9	→ 99
---	---	---	---	---	------

Strategia zachłanna

- Znajdź maksymalną cyfrę:

5	7	3	1	9	→ 9
---	---	---	---	---	-----
- Dołącz ją do poszukiwanej liczby:

5	7	3	1	9	→ 99
---	---	---	---	---	------
- Usuń ją z listy cyfr:

5	7	3	1	→ 99
---	---	---	---	------

Strategia zachłanna

- Znajdź maksymalną cyfrę:

5	7	3	1	9	→ 9
---	---	---	---	---	-----
- Dołącz ją do poszukiwanej liczby:

5	7	3	1	9	→ 99
---	---	---	---	---	------
- Usuń ją z listy cyfr:

5	7	3	1	→ 99
---	---	---	---	------
- Powtarzaj, dopóki na liście znajdują się cyfry.

Strategia zachłanna

- Znajdź maksymalną cyfrę:

5	7	3	1	9	→ 9
---	---	---	---	---	-----
- Dołącz ją do poszukiwanej liczby:

5	7	3	1	9	→ 99
---	---	---	---	---	------
- Usuń ją z listy cyfr:

5	7	3	1	→ 99
---	---	---	---	------
- Powtarzaj, dopóki na liście znajdują się cyfry.
- Znajdź maksymalną cyfrę:

5	7	3	1	→ 99
---	---	---	---	------

Strategia zachłanna

- Znajdź maksymalną cyfrę:

5	7	3	1	9	→ 9
---	---	---	---	---	-----
- Dołącz ją do poszukiwanej liczby:

5	7	3	1	9	→ 99
---	---	---	---	---	------
- Usuń ją z listy cyfr:

5	7	3	1	→ 99
---	---	---	---	------
- Powtarzaj, dopóki na liście znajdują się cyfry.
- Znajdź maksymalną cyfrę:

5	7	3	1	→ 99
---	---	---	---	------
- Dołącz ją do poszukiwanej liczby:

5	7	3	1	→ 997
---	---	---	---	-------

Strategia zachłanna

- Znajdź maksymalną cyfrę:

5	7	3	1	9	→ 9
---	---	---	---	---	-----
- Dołącz ją do poszukiwanej liczby:

5	7	3	1	9	→ 99
---	---	---	---	---	------
- Usuń ją z listy cyfr:

5	7	3	1	→ 99
---	---	---	---	------
- Powtarzaj, dopóki na liście znajdują się cyfry.
- Znajdź maksymalną cyfrę:

5	7	3	1	→ 99
---	---	---	---	------
- Dołącz ją do poszukiwanej liczby:

5	7	3	1	→ 997
---	---	---	---	-------
- Usuń ją z listy cyfr:

5	3	1	→ 997
---	---	---	-------

Strategia zachłanna

- Znajdź maksymalną cyfrę:

5	7	3	1	9	→ 9
---	---	---	---	---	-----
- Dołącz ją do poszukiwanej liczby:

5	7	3	1	9	→ 99
---	---	---	---	---	------
- Usuń ją z listy cyfr:

5	7	3	1	→ 99
---	---	---	---	------
- Powtarzaj, dopóki na liście znajdują się cyfry.
- Znajdź maksymalną cyfrę:

5	7	3	1	→ 99
---	---	---	---	------
- Dołącz ją do poszukiwanej liczby:

5	7	3	1	→ 997
---	---	---	---	-------
- Usuń ją z listy cyfr:

5	3	1	→ 997
---	---	---	-------
- Powtarzaj, dopóki na liście znajdują się cyfry.

Strategia zachłanna

- Znajdź maksymalną cyfrę:

5	7	3	1	9	→ 9
---	---	---	---	---	-----
- Dołącz ją do poszukiwanej liczby:

5	7	3	1	9	→ 99
---	---	---	---	---	------
- Usuń ją z listy cyfr:

5	7	3	1	→ 99
---	---	---	---	------
- Powtarzaj, dopóki na liście znajdują się cyfry.
- Znajdź maksymalną cyfrę:

5	7	3	1	→ 99
---	---	---	---	------
- Dołącz ją do poszukiwanej liczby:

5	7	3	1	→ 997
---	---	---	---	-------
- Usuń ją z listy cyfr:

5	3	1	→ 997
---	---	---	-------
- Powtarzaj, dopóki na liście znajdują się cyfry.
- Znajdź maksymalną cyfrę:

5	3	1	→ 997
---	---	---	-------

Strategia zachłanna

- Znajdź maksymalną cyfrę:

5	7	3	1	9	→ 9
---	---	---	---	---	-----
- Dołącz ją do poszukiwanej liczby:

5	7	3	1	9	→ 99
---	---	---	---	---	------
- Usuń ją z listy cyfr:

5	7	3	1	→ 99
---	---	---	---	------
- Powtarzaj, dopóki na liście znajdują się cyfry.
- Znajdź maksymalną cyfrę:

5	7	3	1	→ 99
---	---	---	---	------
- Dołącz ją do poszukiwanej liczby:

5	7	3	1	→ 997
---	---	---	---	-------
- Usuń ją z listy cyfr:

5	3	1	→ 997
---	---	---	-------
- Powtarzaj, dopóki na liście znajdują się cyfry.
- Znajdź maksymalną cyfrę:

5	3	1	→ 997
---	---	---	-------
- Dołącz ją do poszukiwanej liczby:

5	3	1	→ 9975
---	---	---	--------

Strategia zachłanna

- Znajdź maksymalną cyfrę:

5	7	3	1	9	→ 9
---	---	---	---	---	-----
- Dołącz ją do poszukiwanej liczby:

5	7	3	1	9	→ 99
---	---	---	---	---	------
- Usuń ją z listy cyfr:

5	7	3	1	→ 99
---	---	---	---	------
- Powtarzaj, dopóki na liście znajdują się cyfry.
- Znajdź maksymalną cyfrę:

5	7	3	1	→ 99
---	---	---	---	------
- Dołącz ją do poszukiwanej liczby:

5	7	3	1	→ 997
---	---	---	---	-------
- Usuń ją z listy cyfr:

5	3	1	→ 997
---	---	---	-------
- Powtarzaj, dopóki na liście znajdują się cyfry.
- Znajdź maksymalną cyfrę:

5	3	1	→ 997
---	---	---	-------
- Dołącz ją do poszukiwanej liczby:

5	3	1	→ 9975
---	---	---	--------
- Usuń ją z listy cyfr:

3	1	→ 9975
---	---	--------

Strategia zachłanna

- Znajdź maksymalną cyfrę:

5	7	3	1	9	
---	---	---	---	---	--

 → 9
- Dołącz ją do poszukiwanej liczby:

5	7	3	1	9	
---	---	---	---	---	--

 → 99
- Usuń ją z listy cyfr:

5	7	3	1	
---	---	---	---	--

 → 99
- Powtarzaj, dopóki na liście znajdują się cyfry.
- Znajdź maksymalną cyfrę:

5	7	3	1	
---	---	---	---	--

 → 99
- Dołącz ją do poszukiwanej liczby:

5	7	3	1	
---	---	---	---	--

 → 997
- Usuń ją z listy cyfr:

5	3	1	
---	---	---	--

 → 997
- Powtarzaj, dopóki na liście znajdują się cyfry.
- Znajdź maksymalną cyfrę:

5	3	1	
---	---	---	--

 → 997
- Dołącz ją do poszukiwanej liczby:

5	3	1	
---	---	---	--

 → 9975
- Usuń ją z listy cyfr:

3	1	
---	---	--

 → 9975
- Powtarzaj, dopóki na liście znajdują się cyfry.

Strategia zachłanna

- Znajdź maksymalną cyfrę:

3	1		→ 9975
---	---	--	--------

Strategia zachłanna

- Znajdź maksymalną cyfrę:

3	1		→ 9975
---	---	--	--------
- Dołącz ją do poszukiwanej liczby:

3	1		→ 99753
---	---	--	---------

Strategia zachłanna

- Znajdź maksymalną cyfrę:

3	1		→ 9975
---	---	--	--------
- Dołącz ją do poszukiwanej liczby:

3	1		→ 99753
---	---	--	---------
- Usuń ją z listy cyfr:

1	→ 99753
---	---------

Strategia zachłanna

- Znajdź maksymalną cyfrę:

3	1		→ 9975
---	---	--	--------
- Dołącz ją do poszukiwanej liczby:

3	1		→ 99753
---	---	--	---------
- Usuń ją z listy cyfr:

1	→ 99753
---	---------
- Powtarzaj, dopóki na liście znajdują się cyfry.

Strategia zachłanna

- Znajdź maksymalną cyfrę:

3	1		→ 9975
---	---	--	--------
- Dołącz ją do poszukiwanej liczby:

3	1		→ 99753
---	---	--	---------
- Usuń ją z listy cyfr:

1	→ 99753
---	---------
- Powtarzaj, dopóki na liście znajdują się cyfry.
- Znajdź maksymalną cyfrę:

1	→ 99753
---	---------

Strategia zachłanna

- Znajdź maksymalną cyfrę:

3	1		→ 9975
---	---	--	--------
- Dołącz ją do poszukiwanej liczby:

3	1		→ 99753
---	---	--	---------
- Usuń ją z listy cyfr:

1	→ 99753
---	---------
- Powtarzaj, dopóki na liście znajdują się cyfry.
- Znajdź maksymalną cyfrę:

1	→ 99753
---	---------
- Dołącz ją do poszukiwanej liczby:

1	→ 997531
---	----------

Strategia zachłanna

- Znajdź maksymalną cyfrę:

3	1		→ 9975
---	---	--	--------
- Dołącz ją do poszukiwanej liczby:

3	1		→ 99753
---	---	--	---------
- Usuń ją z listy cyfr:

1	→ 99753
---	---------
- Powtarzaj, dopóki na liście znajdują się cyfry.
- Znajdź maksymalną cyfrę:

1	→ 99753
---	---------
- Dołącz ją do poszukiwanej liczby:

1	→ 997531
---	----------
- Usuń ją z listy cyfr:

→ 997531

Strategia zachłanna

- Znajdź maksymalną cyfrę:

3	1		→ 9975
---	---	--	--------
- Dołącz ją do poszukiwanej liczby:

3	1		→ 99753
---	---	--	---------
- Usuń ją z listy cyfr:

1	→ 99753
---	---------
- Powtarzaj, dopóki na liście znajdują się cyfry.
- Znajdź maksymalną cyfrę:

1	→ 99753
---	---------
- Dołącz ją do poszukiwanej liczby:

1	→ 997531
---	----------
- Usuń ją z listy cyfr:

→ 997531

- Powtarzaj, dopóki na liście znajdują się cyfry.

Strategia zachłanna

- Znajdź maksymalną cyfrę:

3	1		→ 9975
---	---	--	--------
- Dołącz ją do poszukiwanej liczby:

3	1		→ 99753
---	---	--	---------
- Usuń ją z listy cyfr:

1	→ 99753
---	---------
- Powtarzaj, dopóki na liście znajdują się cyfry.
- Znajdź maksymalną cyfrę:

1	→ 99753
---	---------
- Dołącz ją do poszukiwanej liczby:

1	→ 997531
---	----------
- Usuń ją z listy cyfr:

→ 997531

- Powtarzaj, dopóki na liście znajdują się cyfry.
- Wynik: **997531**

Strategia zachłanna

Tankowanie samochodu

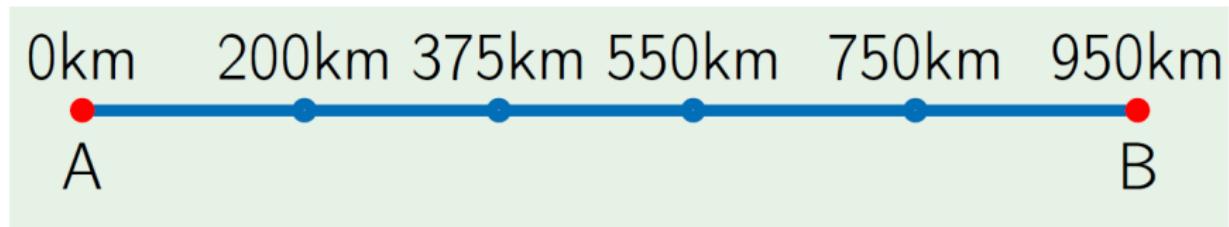
Dystans z pełnym bakiem = 400km



Strategia zachłanna

Tankowanie samochodu

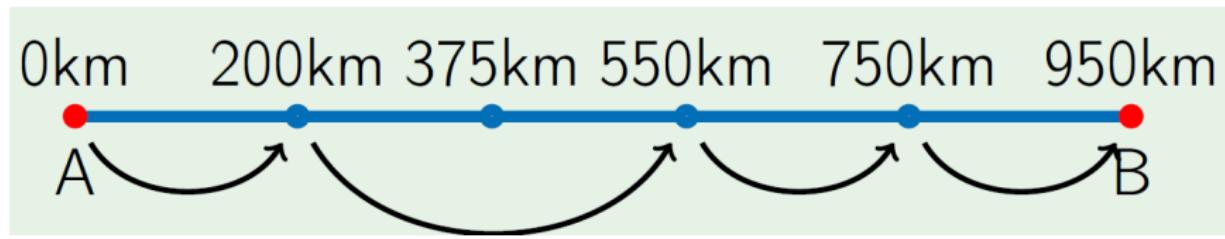
Dystans z pełnym bakiem = 400km



Strategia zachłanna

Tankowanie samochodu

Dystans z pełnym bakiem = 400km



Strategia zachłanna

Tankowanie samochodu

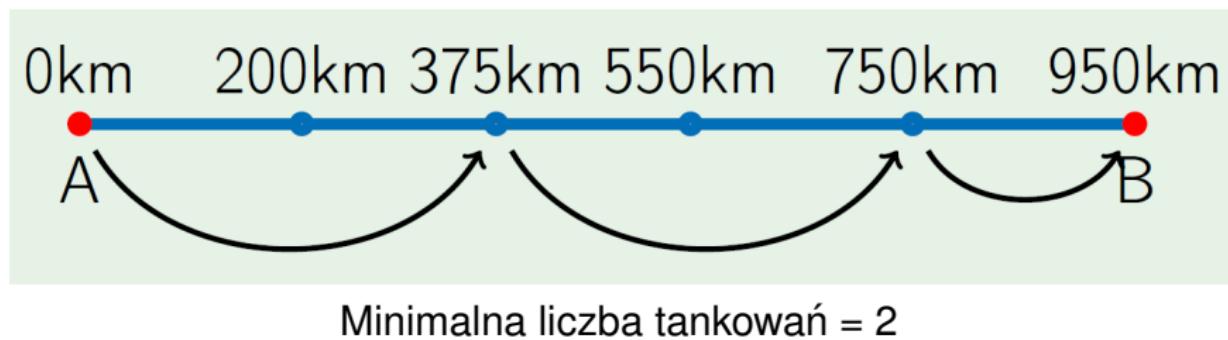
Dystans z pełnym bakiem = 400km



Strategia zachłanna

Tankowanie samochodu

Dystans z pełnym bakiem = 400km



Tankowanie samochodu

Wejście:

Samochód, który może przejechać co najwyżej L kilometrów z pełnym bakiem, punkt początkowy A , punkt docelowy B i n stacji benzynowych w odległościach $x_1 \leq x_2 \leq x_3 \leq \dots \leq x_n$ w kilometrach od A wzdłuż drogi od A do B .

Wyjście:

Minimalna liczba tankowań, wymagana aby dostać się z punktu A do punktu B , oprócz uzupełnienia w punkcie A .

Strategia zachłanna

- Dokonaj wyboru zachłannego
- Zredukuj do mniejszego problemu
- Powtarzaj postępowanie dla mniejszego problemu

Strategia zachłanna

- Dokonaj wyboru zachłannego
- Zredukuj do mniejszego problemu
- Powtarzaj postępowanie dla mniejszego problemu

Która strategię wybrać?

- Uzupełnij paliwo na najbliższej stacji benzynowej
- Uzupełnij paliwo na najdalej dostępnej stacji benzynowej
- Jedź, aż zabraknie paliwa

Strategia zachłanna

- Dokonaj wyboru zachłannego
- Zredukuj do mniejszego problemu
- Powtarzaj postępowanie dla mniejszego problemu

Która strategię wybrać?

- Uzupełnij paliwo na najbliższej stacji benzynowej
- **Uzupełnij paliwo na najdalej dostępnej stacji benzynowej**
- Jedź, aż zabraknie paliwa

Algorytm zachłanny

- Zaczni od A;
- Uzupełnij paliwo na najdalej dostępnej stacji benzynowej G;
- Ustaw G jako nowe A;
- Przejdź z nowego punktu A do punktu B przy minimalnej liczbie uzupełnień;

Algorytm zachłanny

$$A = x_0 \leq x_1 \leq x_2 \leq \dots \leq x_n \leq x_{n+1} = B$$

MinRefills(x , n , L)

```
numRefills ← 0, currentRefill ← 0
while currentRefill ≤ n:
    lastRefill ← currentRefill
    while (currentRefill ≤ n and
           x[currentRefill + 1] - x[lastRefill] ≤ L):
        currentRefill ← currentRefill + 1
    if currentRefill == lastRefill:
        return IMPOSSIBLE
    if currentRefill ≤ n:
        numRefills ← numRefills + 1
return numRefills
```

Algorytm zachłanny

Złożoność

Czas działania algorytmu $\text{MinRefills}(x, n, L)$ wynosi $O(n)$.

Dowód

- `currentRefill` zmienia się od 0 do $n + 1$, co jeden.
- `numRefills` zmienia się od 0 do maksymalnie n , co jeden.
- Zatem mamy $O(n)$ iteracji.

Strategia zachłanna

Podproblem

Podproblem to problem podobny do oryginalnego, ale o mniejszym rozmiarze.

Przykład

- $\text{LargestNumber}(3, 9, 5, 9, 7, 1) =$
„9” + $\text{LargestNumber}(3, 5, 9, 7, 1)$
- Minimalna liczba tankowań z A do B = pierwsze tankowanie w G + minimalna liczba tankowań z G do B

Strategia zachłanna

Podproblem

Podproblem to problem podobny do oryginalnego, ale o mniejszym rozmiarze.

Przykład

- $\text{LargestNumber}(3, 9, 5, 9, 7, 1) =$
„9” + $\text{LargestNumber}(3, 5, 9, 7, 1)$
- Minimalna liczba tankowań z A do B = pierwsze tankowanie w G + minimalna liczba tankowań z G do B

Bezpieczny wybór

Wybór zachłanny nazywany jest **bezpiecznym wyborem**, jeśli istnieje optymalne rozwiązanie zgodne z pierwszym wyborem.

Algorytmy zachłanne

- Algorytmy zachłanne stosowane do rozwiązywania problemów optymalizacyjnych, w których osiągnięcie optymalnego rozwiązania wymaga podejmowania wielu decyzji.
- Algorytm zachłanny przy podejmowaniu jednej z wielu możliwych decyzji zawsze wykonuje działanie, które wydaje się w danej chwili najkorzystniejsze.
- Algorytm zachłanny wybiera lokalnie optymalną możliwość w nadziei, że doprowadzi ona do globalnie optymalnego rozwiązania (działa zachłannie)
- **Algorytmy zachłanne nie zawsze prowadzą do optymalnych rozwiązań.**

Problem wyboru zajęć

- Problem wyboru zajęć, to problem przydzielenia dostępu do zasobu wykorzystywanego podczas wykonywania pewnych zajęć.
- **Założenia:**
 - Niech będzie dany zbiór proponowanych zajęć $S = \{1, \dots, n\}$, do których ma być przydzielona sala wykładowa (zasoby), w której może się odbywać w danej chwili tylko jedno z tych zajęć.
 - Każde zajęcie ma swój czas rozpoczęcia s_i oraz czas zakończenia f_i takie, że $s_i \leq f_i$. Jeżeli zajęcie o numerze i zostanie wytypowane, to zajmuje zasób $[s_i, f_i]$. Zajęcia o numerach i oraz j są zgodne, jeśli $[s_i, f_i] \cap [s_j, f_j] = \emptyset$
- **Problem:** Wyznaczyć największy podzbiór parami zgodnych zajęć.

Jak działa algorytm wyczerpujący (naiwny) ?

- Generujemy wszystkie podzbiory zbioru zajęć
- Wybieramy te podzbiory, które są parami zgodne
- Wybieramy ten podzbiór, który ma najwięcej elementów
- **Obserwacja:** Algorytm wyczerpujący ma złożoność $O(2^n)$

Strategia zachłanna: przykład

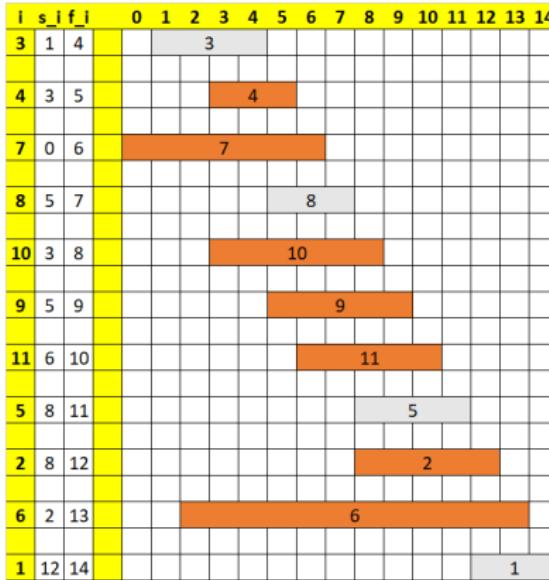
- Źródło:
- Dane są zajęcia:

i	1	2	3	4	5	6	7	8	9	10	11
s_i	12	8	1	3	8	2	0	5	5	3	6
f_i	14	12	4	5	11	13	6	7	9	8	10

- Porządkujemy zajęcia ze względu na czas zakończenia
 $f_1 \leq f_2 \leq \dots \leq f_n$
- Wybieramy na każdym kroku to zajęcie, które ma najwcześniejszy czas zakończenia wśród zajęć, które mogą być dołączone do zbioru - wybór ten maksymalizuje ilość nie zajętego czasu po jego dokonaniu.

Strategia zachłanna: przykład

i	1	2	3	4	5	6	7	8	9	10	11
s_i	12	8	1	3	8	2	0	5	5	3	6
f_i	14	12	4	5	11	13	6	7	9	8	10



Algorytm: GREEDY-ACTIVITY-SELECTOR

- Dane wejściowe stanowią tablice s (czasów rozpoczęcia) oraz f (czasów zakończenia).
- Zakładamy, że zajęcia są posortowane ze względu na czas zakończenia.

GREEDY-ACTIVITY-SELECTOR (s, f)

```
1:  $n := \text{length}[s]$ 
2:  $A := \{1\}$ 
3:  $j := 1$ 
4: for  $i := 2$  to  $n$  do
5:   if  $s_i \geq f_j$  then
6:      $A := A \cup \{i\}$ 
7:      $j := i$ 
8:   end if
9: end for
10: return  $A$ 
```

- Zbiór A zawiera wybrane zajęcia, a zmienna j zawiera numer ostatnio dodanego do A zajęcia.
- Zajęcia są rozpatrywane w porządku rosnącego czasu zakończenia, zatem f_j jest zawsze największym czasem zakończenia zajęcia należącego do A .
- W wierszach 2-3 wybieramy zajęcie 1, zbiór $\{1\}$ staje się wartością zmiennej A , a zmiennej j przypisujemy numer tego zajęcia.
- W wierszach 4-9 rozpatrywane są wszystkie zajęcia; zajęcie zostaje dołączone, jeżeli jest zgodne ze wszystkimi dołączonymi dotychczas zajęciami.
- Aby stwierdzić, czy zajęcie i jest zgodne z każdym zajęciem ze zbioru A , wystarczy sprawdzić czy jego czas rozpoczęcia s_i nie jest wcześniejszy niż czas zakończenia f_j zajęcia ostatnio dodanego do A .
- Jeśli zajęcie i jest zgodne, to w wierszach 6-7 zostaje ono dodane do zbioru A oraz jest aktualizowana

Algorytm: GREEDY-ACTIVITY-SELECTOR - Uwagi

- Zajęcie wybrane przez GREEDY-ACTIVITY-SELECTOR ma zawsze najwcześniejszy czas zakończenia wśród zajęć, które mogą być dołączone bez zakłócenia zgodności zbioru A.
- Wybór jest **zachłanny** w tym sensie, że pozostawia możliwie najwięcej swobody przy wyborze pozostałych zajęć.
- **Twierdzenie:** Algorytm zachłanny generuje rozwiązanie problemu wyboru zajęć o największym rozmiarze.
- Złożoność $O(n)$.

Charakterystyczne cechy problemów poddających się strategii zachłannej

- **optymalna podstruktura** - optymalne rozwiązanie jest funkcją optymalnych rozwiązań podproblemów. Na przykład dla problemu wyboru zajęć własność optymalnej podstruktury polega na tym, że: jeżeli optymalne rozwiązanie A tego problemu rozpoczyna się od zajęć o numerze 1, to $A' = A - \{1\}$ jest optymalnym rozwiązaniem problemu optymalnego wyboru zajęć dla zbioru $S' = \{i \in S : s_i \geq f_1\}$.
- **własność wyboru zachłanego** - za pomocą lokalnie optymalnych (zachłannych) wyborów można uzyskać globalnie optymalne rozwiązanie. Wybory podejmowane w algorytmie zachłannym nie są zależne od wyborów przeszłych. Można formalnie udowodnić (stosując metodę indukcji), że dany problem ma własność wyboru zachłanego.

Problem przydziału zajęć do minimalnej liczby sal

- **Problem:** Dany jest zbiór zajęć, które mają się odbyć w pewnej liczbie sal wykładowych. Należy wyznaczyć taki przydział zajęć do sal, aby liczba użytych sal była najmniejsza.
- **Metoda zachłanna rozwiązująca ten problem:**
 - Niech będzie dany zbiór zajęć $S = 1, \dots, n$, do których ma być przydzielona sala wykładowa (zasoby), w której może się odbywać w danej chwili tylko jedno z tych zajęć.
 - Każde zajęcie ma swój czas rozpoczęcia s_i oraz czas zakończenia f_i takie, że $s_i \leq f_i$. Jeżeli zajęcie o numerze i zostanie wytypowane, to zajmuje zasób $[s_i, f_i]$. Zajęcia o numerach i oraz j są zgodne jeśli $[s_i, f_i] \cap [s_j, f_j] = 0$.
 - Niech zajęcia w S będą uporządkowane niemalejąco ze względu na ich czas rozpoczęcia; Jeżeli znajdą się dwa zajęcia i oraz j takie, że $s_i = s_j$, to wówczas pierwsze będzie to zajęcie, które ma dłuższy czas trwania.

Algorytm: GREEDY-ROOM-SELECTOR

- Dane wejściowe stanowią tablice s (czasów rozpoczęcia) oraz f (czasów zakończenia).
- Zakładamy, że zajęcia są posortowane ze względu na czas rozpoczęcia.

GREEDY-ROOM-SELECTOR (s, f)

```
1:  $k := 1$ 
2:  $A := \{1, 2, \dots, \text{length}[s]\}$ 
3: while  $A \neq \emptyset$  do
4:    $p[k] := \text{GREEDY-SELECTOR}(s, f)$ 
5:    $A := A - p[k]$ 
6:   if  $A \neq \emptyset$  then
7:      $k := k + 1$ 
8:   end if
9: end while
10: return  $k$ 
```

- Zadaniem procedury pomocniczej GREEDY-SELECTOR jest wybór ze zbioru S podzbioru zawierającego wszystkie parami zgodne zajęcia, wśród których jest zajęcie nr 1.
- Działanie tej procedury jest analogiczne jak działanie procedury GREEDY-ACTIVITY-SELECTOR. Różnica polega jedynie na tym, że GREEDY-SELECTOR pobiera zajęcia uporządkowane zgodnie z czasem ich rozpoczęcia, a GREEDY-ACTIVITY-SELECTOR pobiera zajęcia uporządkowane zgodnie z czasem ich zakończenia.
- Złożoność $O(n^2)$

Algorytm: GREEDY-ROOM-SELECTOR - Uwagi

- Liczba sal wyznaczona GREEDY-ROOM-SELECTION jest zawsze najmniejsza
- Wybór jest zachłanny w tym sensie, że pozostawia w danej Sali minimalną ilość niewykorzystanego czasu
- **Twierdzenie:** Algorytm GREEDY-ROOM-SELECTION generuje optymalne rozwiązanie problemu przydziału zajęć do jak najmniejszej liczby sal.

Problem wydawania reszty za pomocą jak najmniejszej liczby monet

- **Problem:** Mamy resztę R i N monet o nominałach o określonych wartościach całkowitych. Monet o określonym nominale jest nieograniczona liczba. Należy podać jak najmniejszą ilość monet potrzebną do wydania reszty.
- **Przykład:** Wydać 98 PLN, dysponując nominałami:

50, 20, 10, 5, 2, 1.

- **Rozwiązanie:** Porządkujemy nominały malejąco ze względu na ich wartości. Wydajemy następująco:

- 50 - pozostało 48
- 20 - pozostało 28
- 20 - pozostało 8
- 5 - pozostało 3
- 2 - pozostało 1
- 1 - pozostało 0

Algorytm: GREEDY-GIVE-CHANGE

- **Założenie:** c jest uporządkowaną malejąco tablicą nominałów, sum jest resztą do wydania, k jest liczbą nominałów różnego rodzaju.

GREEDY-GIVE-CHANGE (sum, c, k)

```
1: for  $i := 0$  to  $k$  do
2:   if  $(sum \text{ div } c[i]) \neq 0$  then
3:      $change[i] := sum \text{ div } c[i]$ 
4:      $sum := sum \text{ mod } c[i]$ 
5:   else
6:      $change[i] := 0$ 
7:   end if
8: end for
9: return  $change$ 
```

- Do tablicy $change$ wpisujemy największą możliwą liczbę nominału $c^k > c^{k-1} > \dots > c^0$.
- Jeżeli jakiegoś nominału nie można użyć do wydania określonej reszty, tzn. nie jest spełniony warunek $(sum \text{ div } c[i])! = 0$, to do tablicy $change$ wpisywana jest wartość zero - wiersz 6 algorytmu.
- Złożoność $O(n)$

Problem wydawania reszt - uwagi

- Istnieją zbiory nominałów, dla których podany algorytm nie daje optymalnego rozwiązania.
- **Przykład 1:** Wydać 10PLN za pomocą nominałów 1, 5 i 6.
 - Rozwiązanie algorytmu zachłannego: $6 + 1 + 1 + 1 + 1$, czyli 5 monet.
 - Rozwiązanie optymalne: $5 + 5$
- **Przykład 2:** Wydać 66 PLN za pomocą nominałów 50, 20 i 1.
 - Rozwiązanie algorytmu zachłannego: $50 + 16 * 1$, czyli 17 nominałów.
 - Rozwiązanie optymalne: $3 * 20 + 6 * 1$, czyli 9 nominałów.