

Wstęp do programowania

INP001213Wcl

rok akademicki 2018/19

semestr zimowy

Wykład 9

Karol Tarnowski

karol.tarnowski@pwr.edu.pl

A-1 p. 411B



Plan prezentacji

- Zasada dziel i zwyciężaj
- Przykłady
 - znajdowanie maksimum i minimum
 - przeszukiwanie binarne
 - metoda bisekcji (połowienia)
 - sortowanie przez scalanie
 - sortowanie szybkie
 - wypełnianie planszy

Na podstawie:

- M. M. Sysło, *Algorytmy*

Zasada dziel i zwyciężaj

Jednoczesne znajdowanie maksimum i minimum

- podziel problem na podproblemy

podziel zbiór na dwa podzbiory: kandydatów na maksimum i kandydatów na minimum

- znajdź rozwiązania podproblemów

znajdź minimum i maksimum w zbiorach kandydatów

- połącz rozwiązania podproblemów w rozwiązanie głównego problemu

zwróć wynik

min	3		1			4		6
	3	5	1	2	8	4	7	6
max		5		2	8		7	



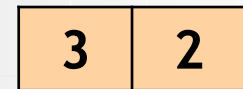
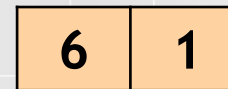
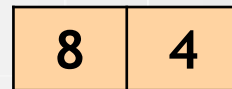
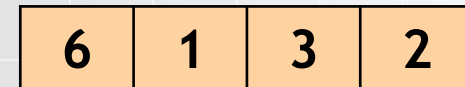
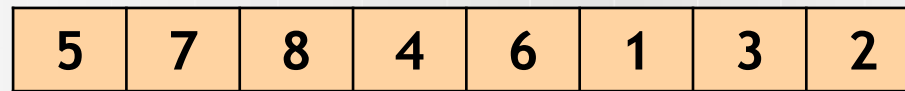
Zasada dziel i zwyciężaj

- podziel problem na co najmniej 2 podproblemy (takie same lub bardzo podobne)
- znajdź rozwiązania podproblemów
- połącz rozwiązania podproblemów w rozwiązanie głównego problemu

- najczęściej podproblemy są tożsame z problemem głównym, ale dla mniejszego zbioru danych

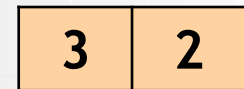
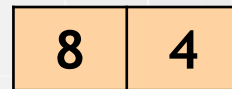
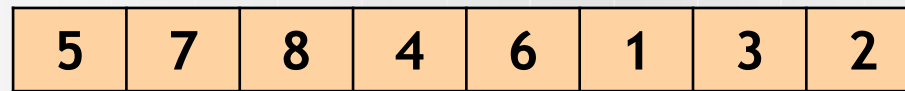
Zasada dziel i zwyciężaj

Jednoczesne znajdowanie maksimum i minimum



Zasada dziel i zwyciężaj

Jednoczesne znajdowanie maksimum i minimum





Przeszukiwanie binarne

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
---	---	---	---	---	---	---	---	---	----	----	----	----	----	----	----	----	----	----	----



1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
---	---	---	---	---	---	---	---	---	----	----	----	----	----	----	----	----	----	----	----

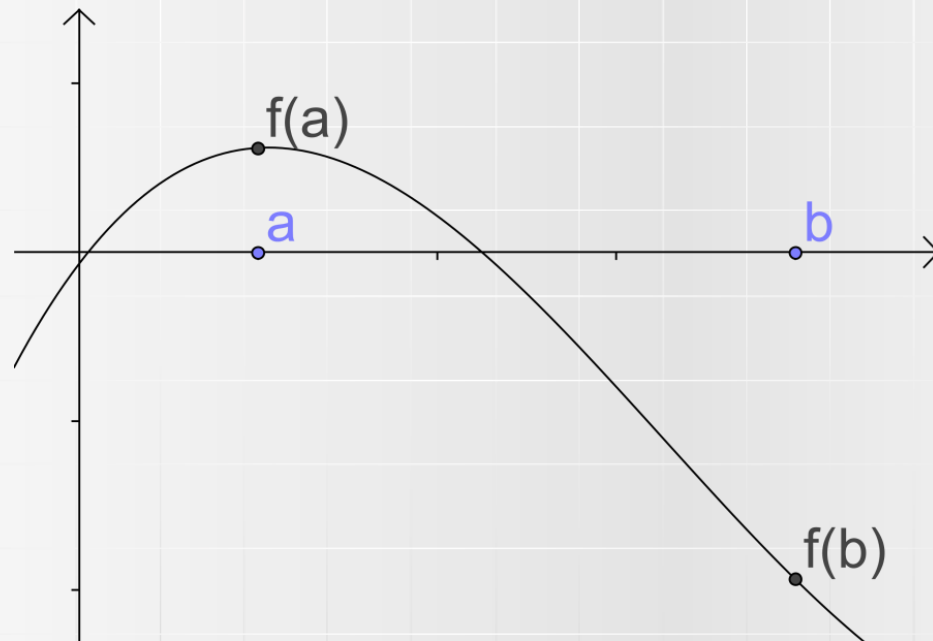


1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
---	---	---	---	---	---	---	---	---	----	----	----	----	----	----	----	----	----	----	----



Metoda bisekcji

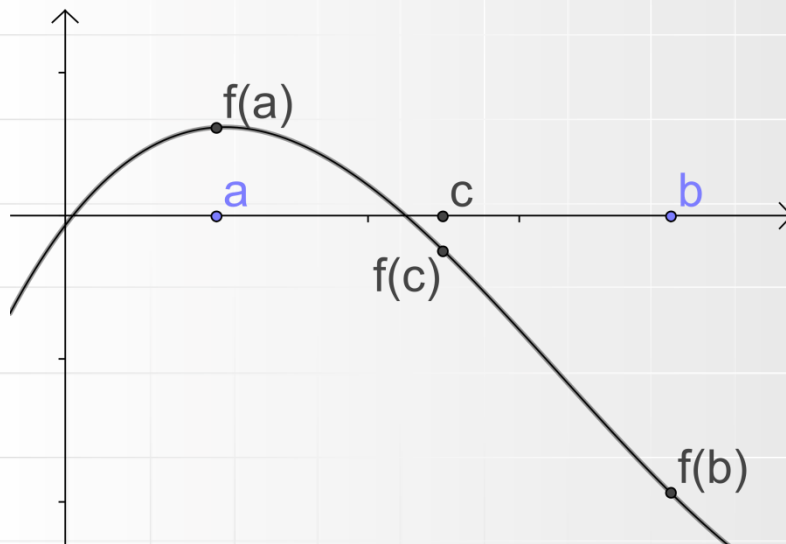
- jeśli f jest funkcją ciągłą w przedziale $[a, b]$ i jeśli $f(a)f(b) < 0$, to funkcja ta musi mieć zero w (a, b)



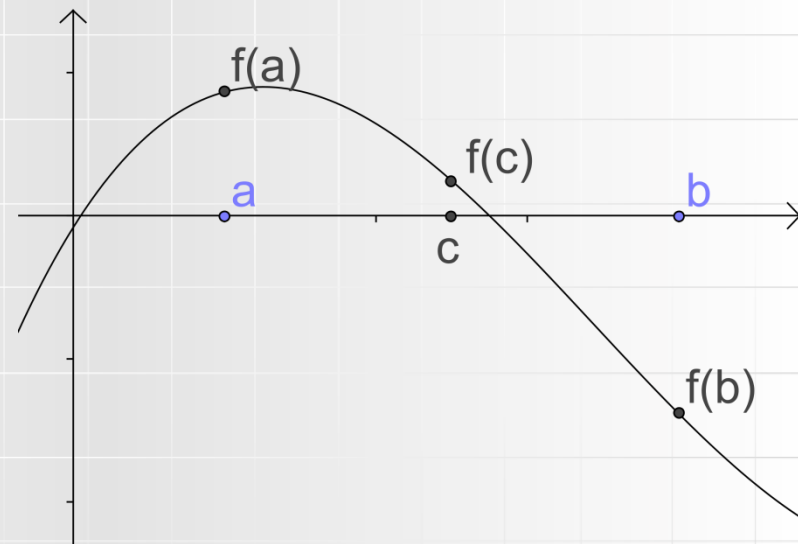


Metoda bisekcji

wyznaczamy punkt $c = \frac{1}{2}(a + b)$ oraz
wartość funkcji $f(c)$



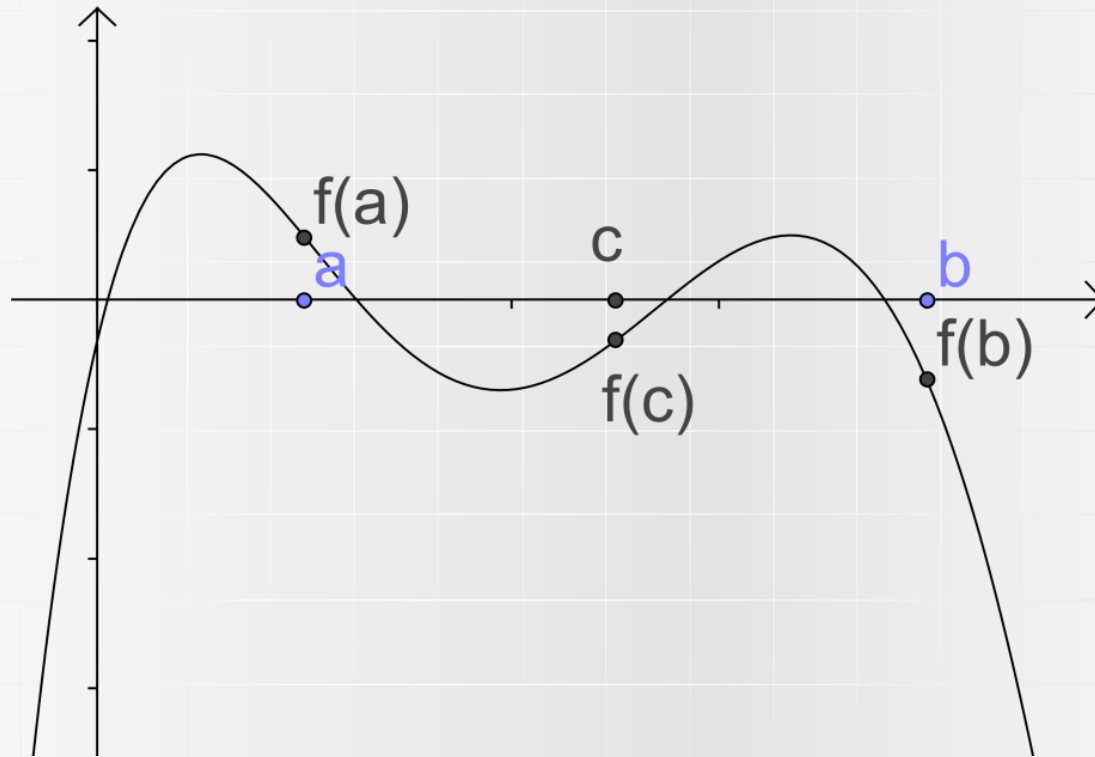
jeśli $f(a)f(c) < 0$ to $b \leftarrow c$



jeśli $f(b)f(c) < 0$ to $a \leftarrow c$



Metoda bisekcji



jeśli $f(a)f(c) < 0$ to $b \leftarrow c$



Metoda bisekcji

Kryteria zakończenia:

- przekroczenie maksymalnej liczby kroków
- zadowalająco mały błąd
- zadowalająco mała wartość funkcji

Algorytmy sortowania

Sortowanie przez scalanie

- podziel tablicę na dwie równe (prawie równe) części
- zastosuj sortowanie przez scalanie do każdej z nich oddzielnie
- połącz posortowane podciągi w jeden ciąg posortowany

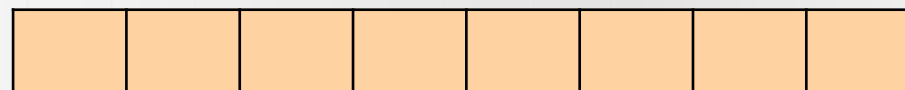
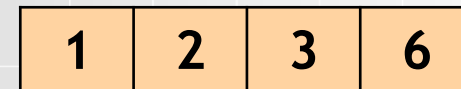
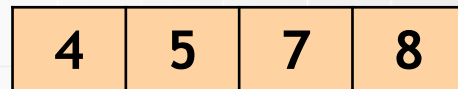
Algorytmy sortowania

Sortowanie przez scalanie

- Procedura scalania dwóch ciągów $a[0..n-1]$ i $b[0..m-1]$ do ciągu $c[0..n+m-1]$
 1. ustaw indeksy na początek ciągów: $i=0$, $j=0$
 2. jeśli w ciągu a nie pozostało już nic do przetworzenia ($i \geq n$), to dołącz pozostałe elementy z b do c i zakończ
 3. jeśli w ciągu b nie pozostało już nic do przetworzenia ($j \geq m$), to dołącz pozostałe elementy z a do c i zakończ
 4. jeśli $a[i] \leq b[j]$ to dołącz $a[i]$ do c i zwiększ i o 1, w p.p. dołącz $b[j]$ do c i zwiększ j o 1.
 5. powtarzaj od 2.

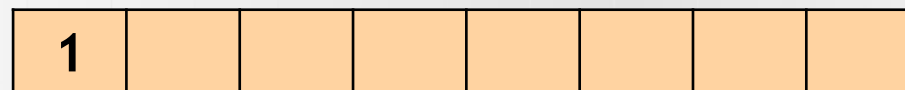
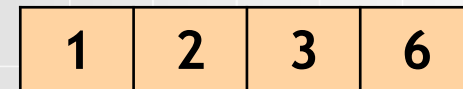
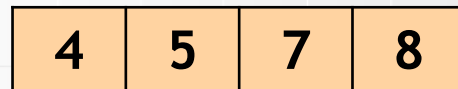
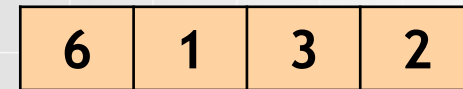
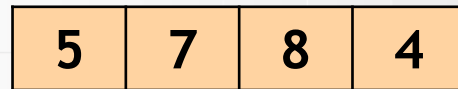
Algorytmy sortowania

Sortowanie przez scalanie



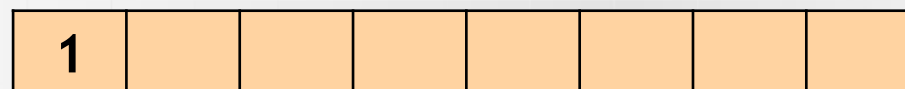
Algorytmy sortowania

Sortowanie przez scalanie



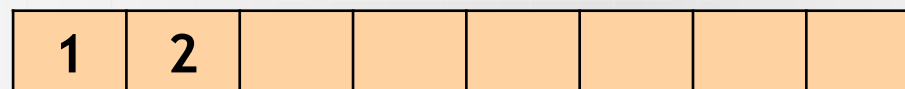
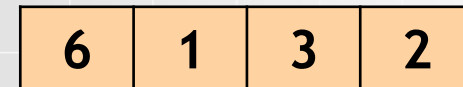
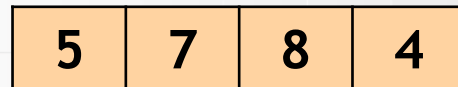
Algorytmy sortowania

Sortowanie przez scalanie



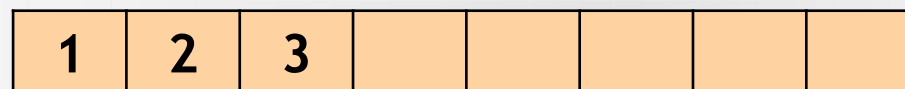
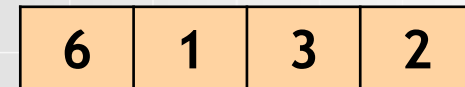
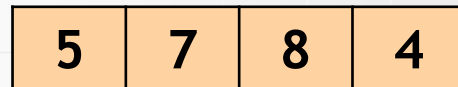
Algorytmy sortowania

Sortowanie przez scalanie



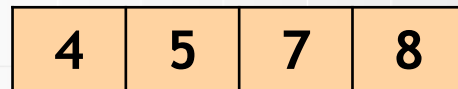
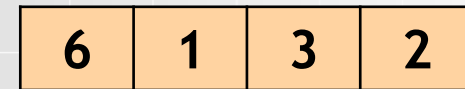
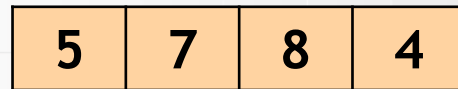
Algorytmy sortowania

Sortowanie przez scalanie



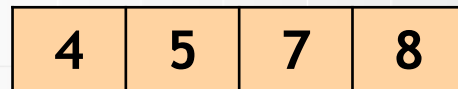
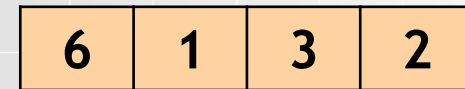
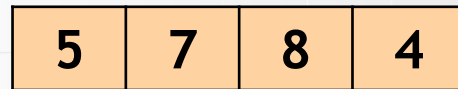
Algorytmy sortowania

Sortowanie przez scalanie



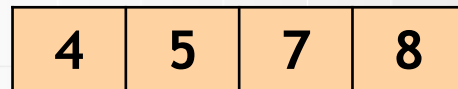
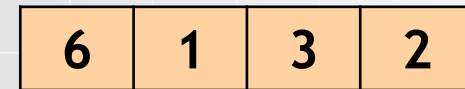
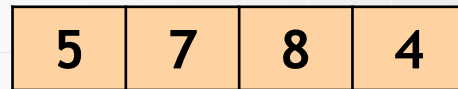
Algorytmy sortowania

Sortowanie przez scalanie



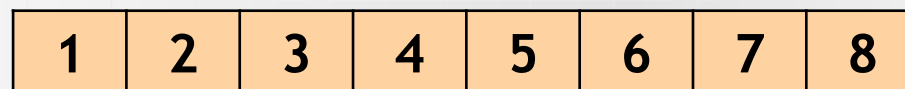
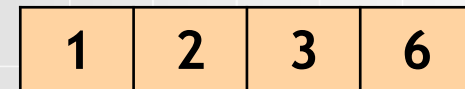
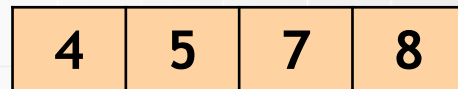
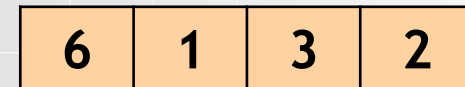
Algorytmy sortowania

Sortowanie przez scalanie



Algorytmy sortowania

Sortowanie przez scalanie



Algorytmy sortowania

Sortowanie przez scalanie

- Dane: ciąg liczb x_p, x_{p+1}, \dots, x_k
 - Wynik: uporządkowanie ciągu liczb od najmniejszej do największej
1. jeśli $p < k$ to $s = (p + k)/2$ w p. p. ciąg jest posortowany
 2. zastosuj ten algorytm do ciągu x_p, \dots, x_s
 3. zastosuj ten algorytm do ciągu x_{s+1}, \dots, x_k
 4. scal ciągi x_p, \dots, x_s oraz x_{s+1}, \dots, x_k i umieść w ciągu x_p, x_{p+1}, \dots, x_k

Algorytmy sortowania

Sortowanie szybkie

- Podziel tablicę na dwie części: mniejsze i większe od wybranego elementu
- Zastosuj sortowanie szybkie do każdej z nich oddzielnie
- Połącz posortowane podciągi w jeden ciąg posortowany

Algorytmy sortowania

Sortowanie szybkie

5	7	8	4	6	1	3	2
---	---	---	---	---	---	---	---

2	7	8	4	6	1	3	5
---	---	---	---	---	---	---	---



2	7	8	4	6	1	3	5
---	---	---	---	---	---	---	---



Algorytmy sortowania

Sortowanie szybkie

5	7	8	4	6	1	3	2
---	---	---	---	---	---	---	---

2	7	8	4	6	1	3	5
---	---	---	---	---	---	---	---



2	7	8	4	6	1	3	5
---	---	---	---	---	---	---	---



Algorytmy sortowania

Sortowanie szybkie

5	7	8	4	6	1	3	2
---	---	---	---	---	---	---	---

2	7	8	4	6	1	3	5
---	---	---	---	---	---	---	---



2	7	8	4	6	1	3	5
---	---	---	---	---	---	---	---



Algorytmy sortowania

Sortowanie szybkie

5	7	8	4	6	1	3	2
---	---	---	---	---	---	---	---

2	7	8	4	6	1	3	5
---	---	---	---	---	---	---	---



2	7	8	4	6	1	3	5
---	---	---	---	---	---	---	---



Algorytmy sortowania

Sortowanie szybkie

5	7	8	4	6	1	3	2
---	---	---	---	---	---	---	---

2	7	8	4	6	1	3	5
---	---	---	---	---	---	---	---



2	4	8	7	6	1	3	5
---	---	---	---	---	---	---	---



Algorytmy sortowania

Sortowanie szybkie

5	7	8	4	6	1	3	2
---	---	---	---	---	---	---	---

2	7	8	4	6	1	3	5
---	---	---	---	---	---	---	---



2	4	8	7	6	1	3	5
---	---	---	---	---	---	---	---



Algorytmy sortowania

Sortowanie szybkie

5	7	8	4	6	1	3	2
---	---	---	---	---	---	---	---

2	7	8	4	6	1	3	5
---	---	---	---	---	---	---	---



2	4	8	7	6	1	3	5
---	---	---	---	---	---	---	---



Algorytmy sortowania

Sortowanie szybkie

5	7	8	4	6	1	3	2
---	---	---	---	---	---	---	---

2	7	8	4	6	1	3	5
---	---	---	---	---	---	---	---



2	4	1	7	6	8	3	5
---	---	---	---	---	---	---	---



Algorytmy sortowania

Sortowanie szybkie

5	7	8	4	6	1	3	2
---	---	---	---	---	---	---	---

2	7	8	4	6	1	3	5
---	---	---	---	---	---	---	---



2	4	1	7	6	8	3	5
---	---	---	---	---	---	---	---



Algorytmy sortowania

Sortowanie szybkie

5	7	8	4	6	1	3	2
---	---	---	---	---	---	---	---

2	7	8	4	6	1	3	5
---	---	---	---	---	---	---	---



2	4	1	3	6	8	7	5
---	---	---	---	---	---	---	---



Algorytmy sortowania

Sortowanie szybkie

5	7	8	4	6	1	3	2
---	---	---	---	---	---	---	---

2	7	8	4	6	1	3	5
---	---	---	---	---	---	---	---



2	4	1	3	6	8	7	5
---	---	---	---	---	---	---	---



Algorytmy sortowania

Sortowanie szybkie

5	7	8	4	6	1	3	2
---	---	---	---	---	---	---	---

2	7	8	4	6	1	3	5
---	---	---	---	---	---	---	---



2	4	1	3	5	8	7	6
---	---	---	---	---	---	---	---

Algorytmy sortowania

Sortowanie szybkie

5	7	8	4	6	1	3	2
---	---	---	---	---	---	---	---

2	7	8	4	6	1	3	5
---	---	---	---	---	---	---	---



2	4	1	3	5	8	7	6
---	---	---	---	---	---	---	---

1	2	3	4	5	8	7	6
---	---	---	---	---	---	---	---

1	2	3	4	5	8	7	6
---	---	---	---	---	---	---	---

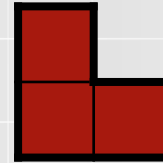
1	2	3	4	5	8	7	6
---	---	---	---	---	---	---	---

1	2	3	4	5	6	7	8
---	---	---	---	---	---	---	---

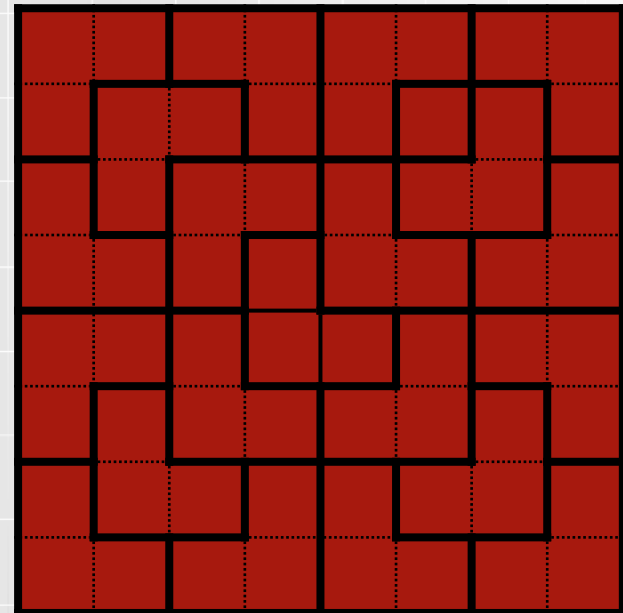
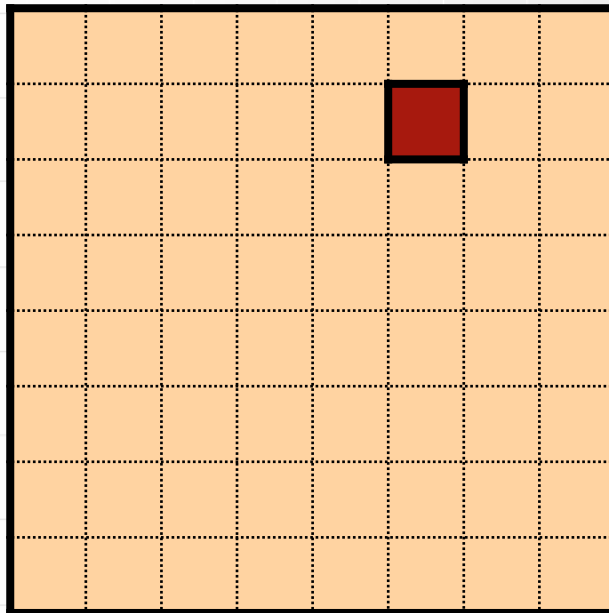


Wypełnianie planszy

- Dysponując klockami

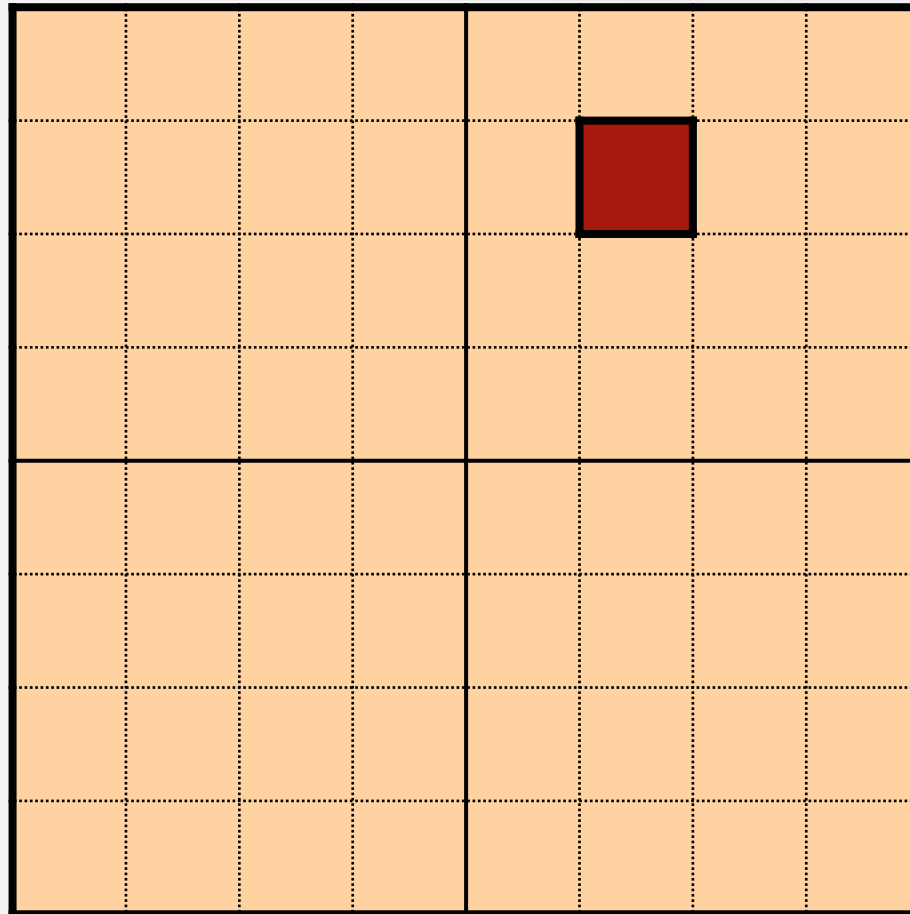


oraz planszą $2^n \times 2^n$ z brakującym polem



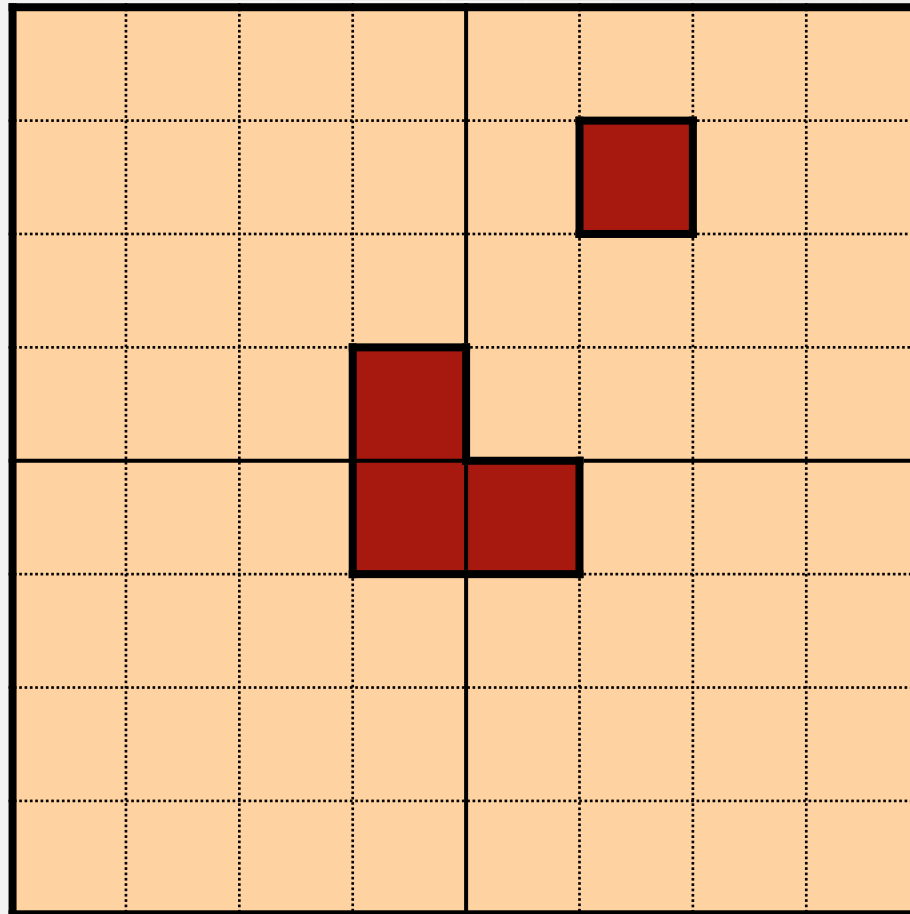


Wypełnianie planszy



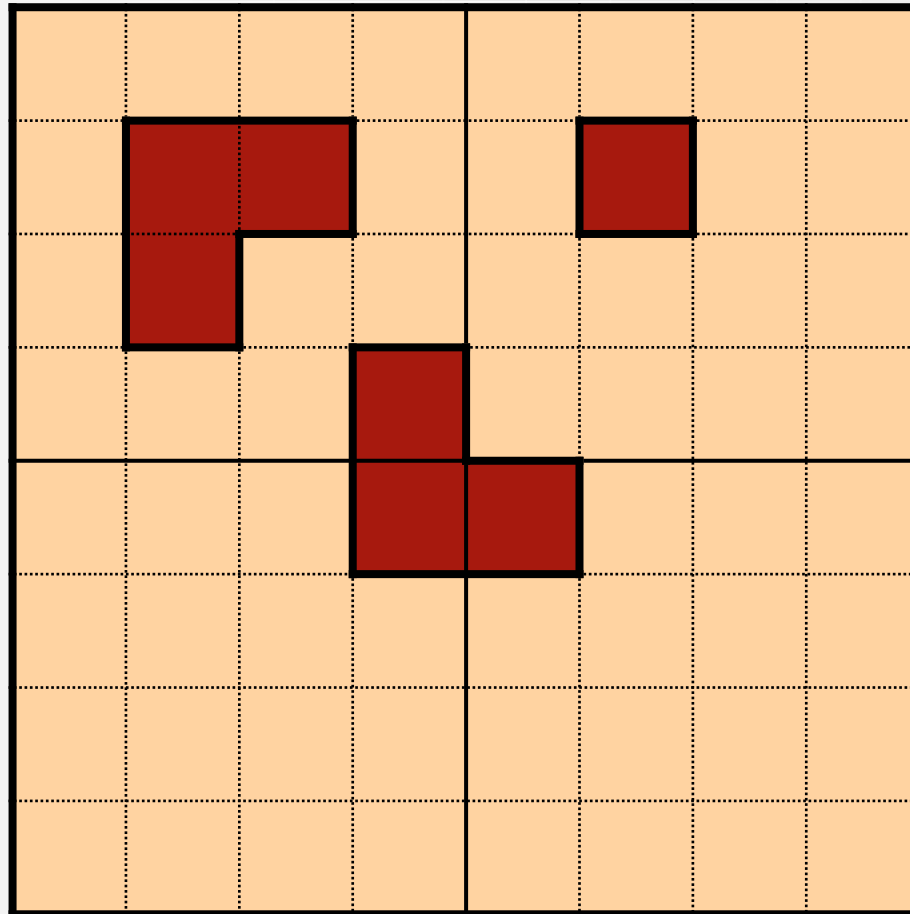


Wypełnianie planszy

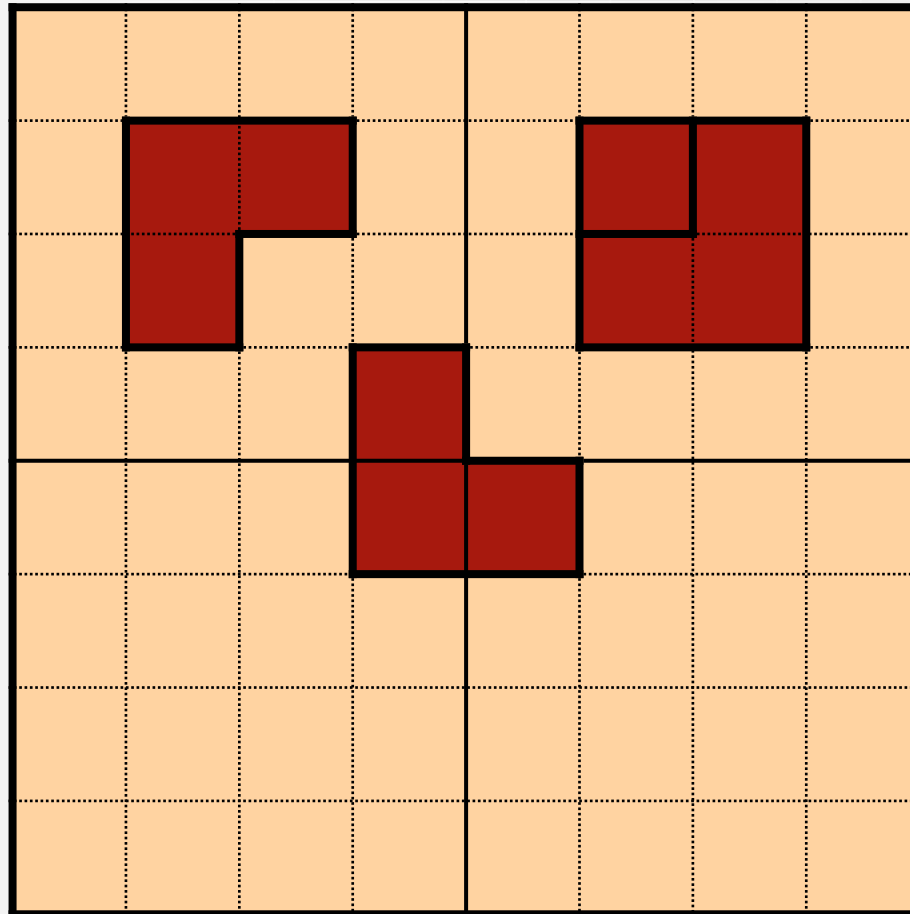




Wypełnianie planszy

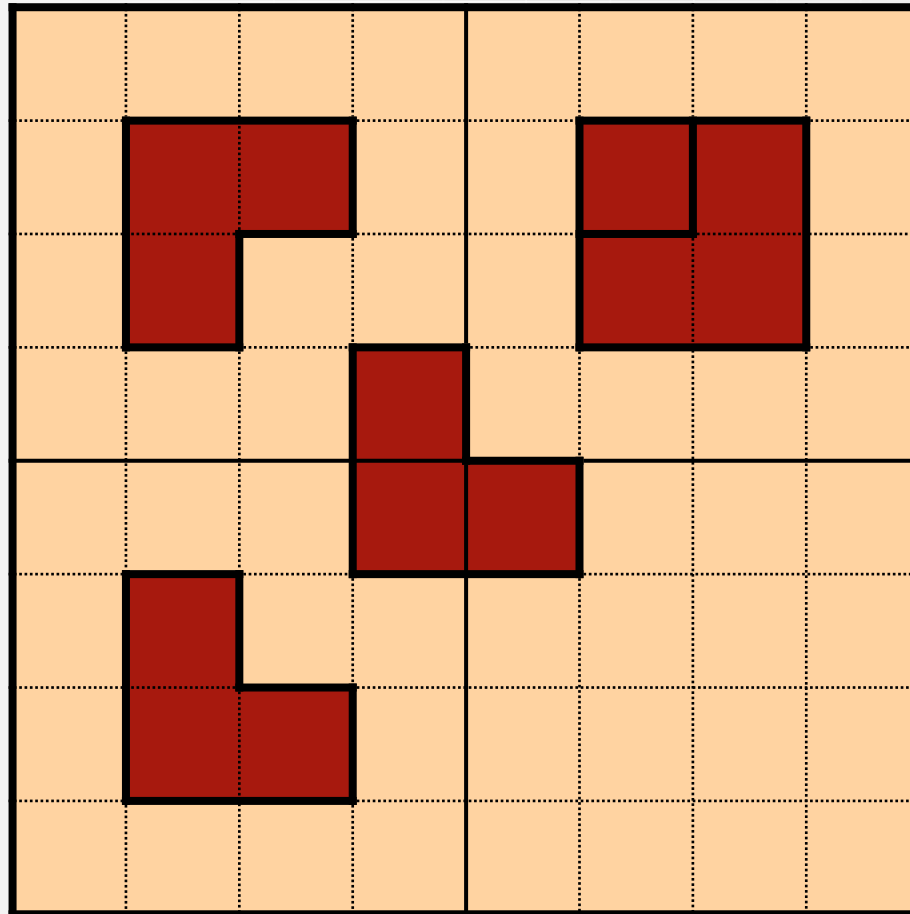


Wypełnianie planszy

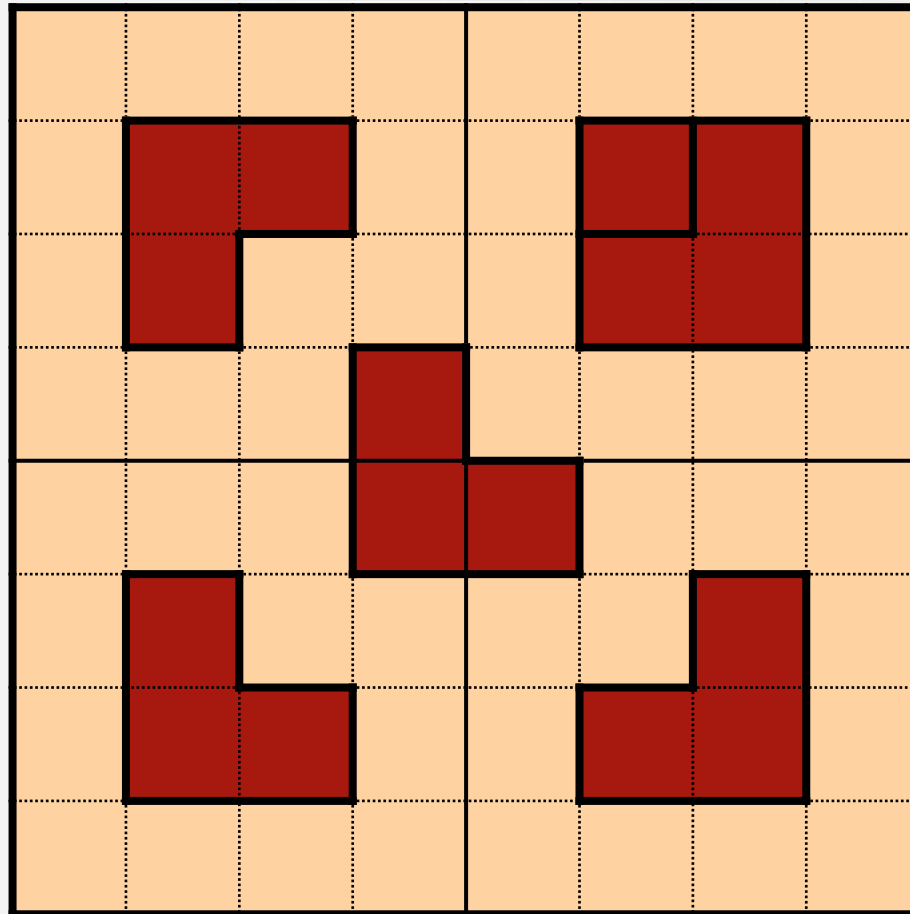




Wypełnianie planszy

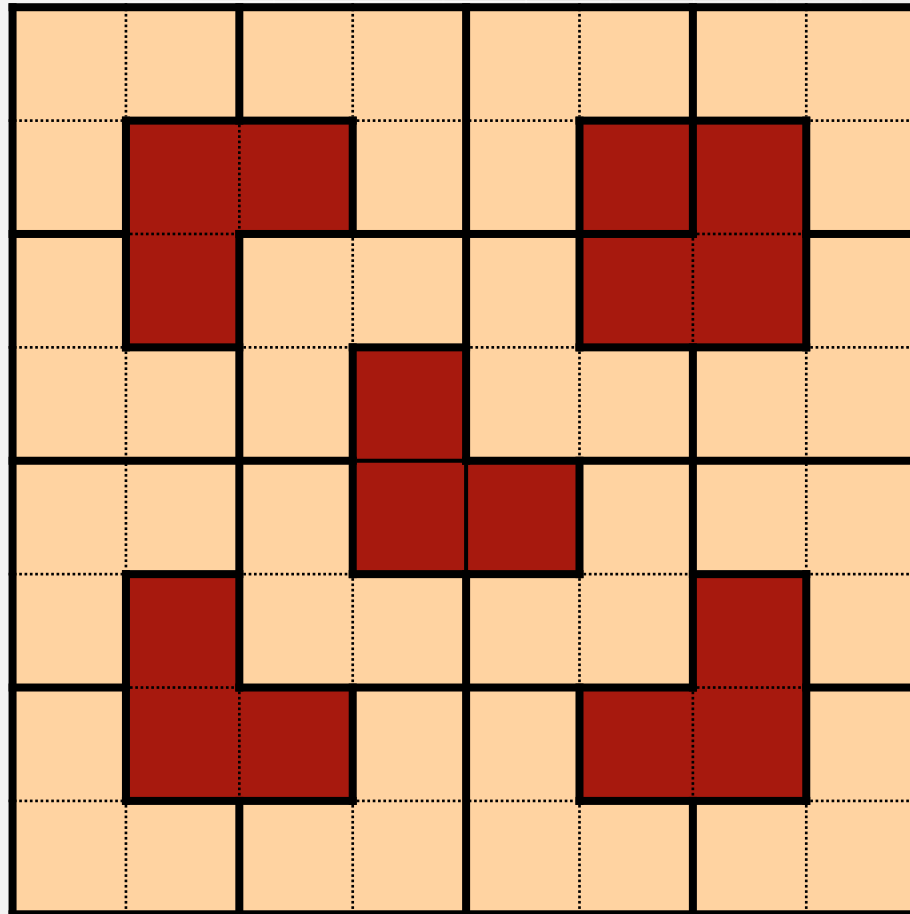


Wypełnianie planszy





Wypełnianie planszy





Podsumowanie

- Zasada dziel i zwyciężaj
 - podziel problem na co najmniej 2 podproblemy (takie same lub bardzo podobne)
 - znajdź rozwiązania podproblemów
 - połącz rozwiązania podproblemów w rozwiązanie głównego problemu
- Przykłady algorytmów