

Sprawozdanie 5

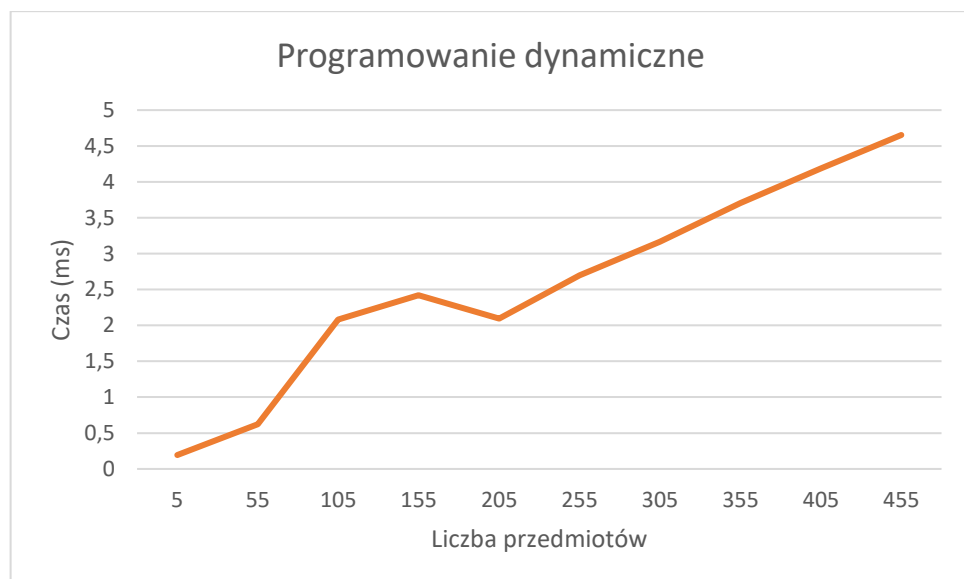
Programowanie dynamiczne

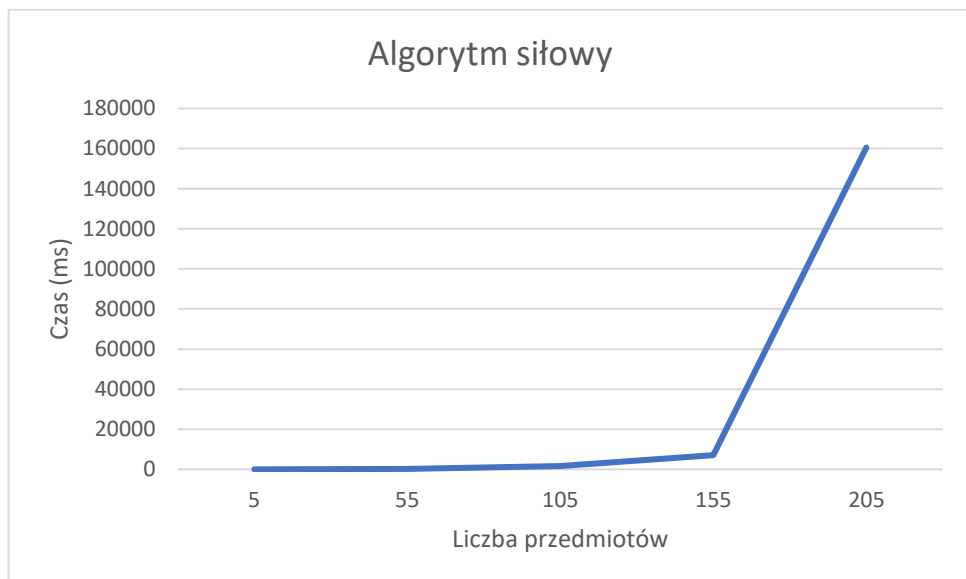
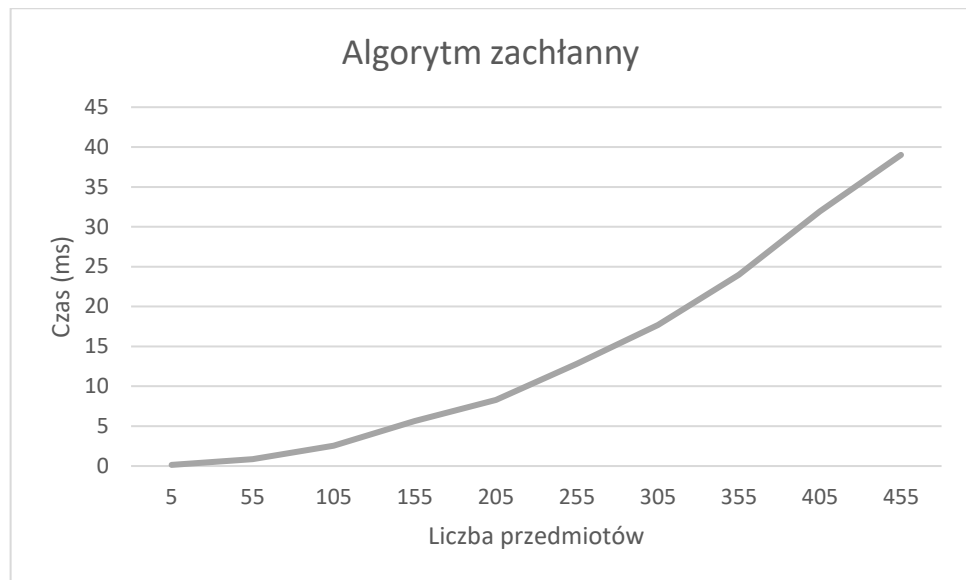
Andrei Staravoitau, nr 150218,

Informatyka I6, II semestr

- 1. Problem plecakowy** - jeden z najczęściej poruszanych problemów optymalizacyjnych. Nazwa zagadnienia pochodzi od maksymalizacyjnego problemu wyboru przedmiotów, tak by ich wartość sumaryczna była jak największa i jednocześnie mieściły się w plecaku. Przy podanym zbiorze elementów o podanej wadze i wartości należy wybrać taki podzbiór, by suma wartości była możliwie jak największa, a suma wag była nie większa od danej pojemności plecaka.
- 2. Optymalizacyjna wersja problemu plecakowego** odnosi się do klasy problemów **NP-trudnych** (w zwykłym sensie).
Decyzyjna wersja problemu plecakowego odnosi się do klasy problemów **NP-zupełnych**.
- 3.** Dla rozwiązania problemu plecakowego użyłem 3 algorytmy: programowania dynamicznego ($O(n \cdot c)$), zachłanny ($O(n \cdot \log_2 n)$) i siłowy ($O(2^n)$).

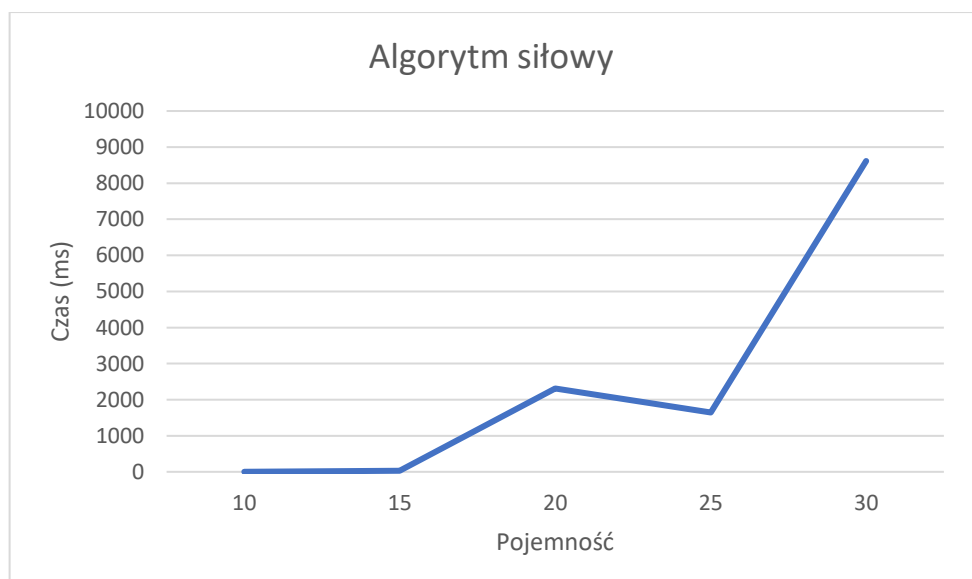
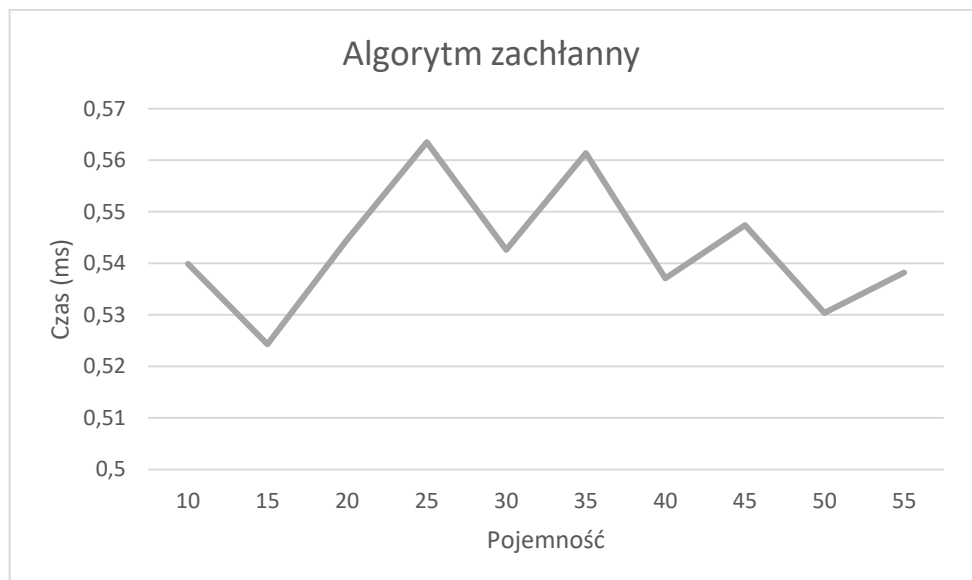
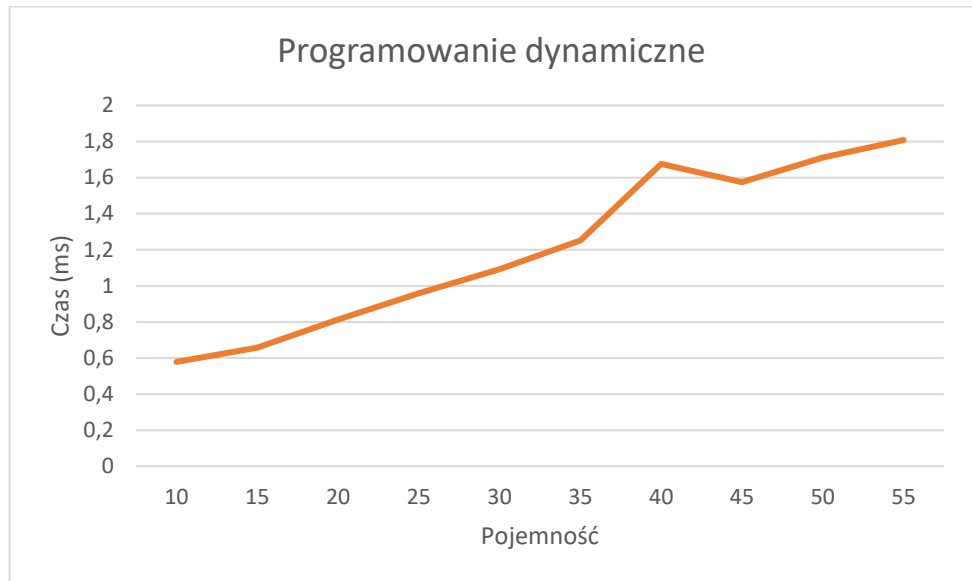
Wykresy (stała pojemność)





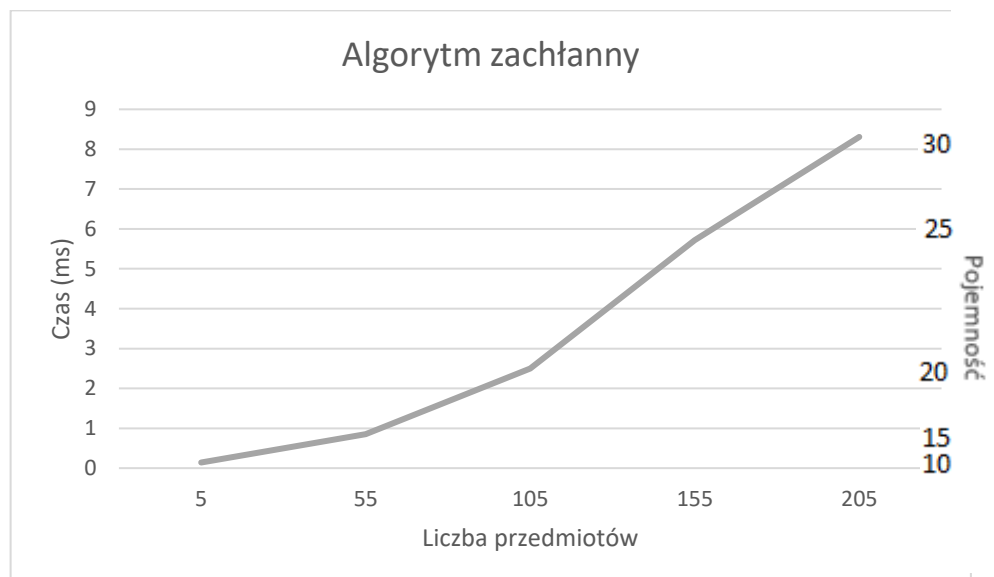
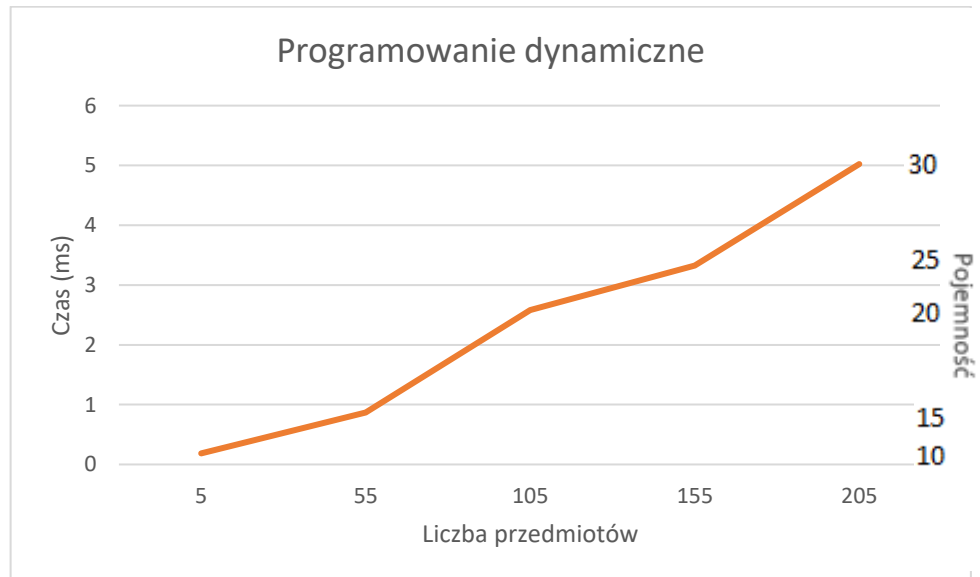
Zgodnie z przedstawionymi wykresami algorytm programowania dynamicznego jest najlepszym i najszybszym z wybranych algorytmów. W każdym razie czas wykonania rośnie razem ze wzrostem liczby przedmiotów.

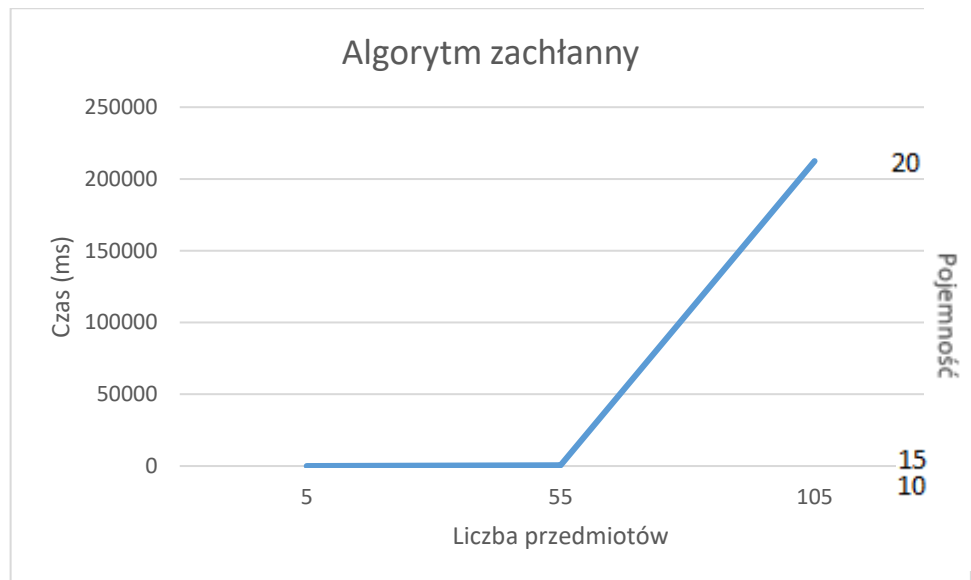
Wykresy (stała liczba przedmiotów)



Zgodnie z przedstawionymi wykresami czas wykonania algorytmu zachłannego jest niezależny od pojemności plecaka. W pozostałych przypadkach czas wykonania rośnie razem ze wzrostem pojemności.

Wykresy ($t=f(n,b)$)





Code

```
import copy
import random
from timeit import default_timer as timer

def wpis():
    n, b = 0, 0
    lista = []
    while n <= 0 or b <= 0:
        while True:
            try:
                n, b = map(int, input("Proszę podać liczbę przedmiotów i pojemność plecaka ").split())
                break
            except ValueError:
                print("Nieprawidłowy typ danych")
    for i in range(n):
        while True:
            try:
                lista.append(list(map(int, input("Proszę podać rozmiar i wartość przedmiotu ").split()))
                if len(lista[i]) == 2 and lista[i][0] > 0 and lista[i][1] > 0:
                    break
                print("Nieprawidłowy typ danych")
                lista.pop()
            except ValueError:
                print("Nieprawidłowy typ danych")
    return n, b, lista

def file():
    file = open("file.txt", 'r')
    l = [line.split() for line in file]
    for i in range(len(l)):
        for j in range(len(l[i])):
            l[i][j] = int(l[i][j])
    file.close()
    n, b = l[0][0], l[0][1]
    for i in range(1, len(l)):
        lista.append(l[i])
    return n, b
```

```

def rand():
    for i in range(n):
        lista.append([])
        for j in range(2):
            lista[i].append(random.randint(1,10))

def AD():
    lista_AD = copy.deepcopy(lista)
    macierz = [[0 for j in range(b+1)] for i in range(n+1)]
    for i in range(n+1):
        for j in range(b+1):
            if i==0 or b==0:
                macierz[i][j] = 0
            elif lista_AD[i-1][0]<=j:
                macierz[i][j]=max(lista_AD[i-1][1]+macierz[i-1][j-lista_AD[i-1][0]], macierz[i-1][j])
            else:
                macierz[i][j]=macierz[i-1][j]

    check=macierz[n][b]
    i=n
    j=b
    rozmiar = 0
    elementy=[]
    while check!=0 and i>0 and j>0:
        if macierz[i][j]>macierz[i-1][j]:
            elementy.append(i)
            i-=1
            j-=lista_AD[i][0]
            check -= lista_AD[i][1]
            rozmiar += lista_AD[i][0]
        else:
            i-=1
    print("Elementy w plecaku: ", elementy,"nRozmiar elementów: ", rozmiar,
        "nWartość elementów: ",macierz[n][b])

def AZ():
    lista_w_na_r = copy.deepcopy(lista)
    wartosc = 0
    rozmiar = 0
    e = 0
    elementy = []
    for i in range(n):
        for j in range(0,n-i-1):
            if
lista_w_na_r[j][0]/lista_w_na_r[j][1]>lista_w_na_r[j+1][0]/lista_w_na_r[j+1][1]:
                lista_w_na_r[j], lista_w_na_r[j+1] = lista_w_na_r[j+1],
lista_w_na_r[j]
        while rozmiar<b and e<n:
            if rozmiar+lista_w_na_r[e][0]<=b:
                wartosc += lista_w_na_r[e][1]
                rozmiar+=lista_w_na_r[e][0]
                elementy.append(lista.index(lista_w_na_r[e])+1)
            e+=1
    print("Elementy w plecaku: ", elementy,"nRozmiar elementów: ", rozmiar,
        "nWartość elementów: ",wartosc)

def silowy():
    pomoc = []
    wynik =[]

```

```

max_w = 0
max_r = 0
for i in range(pow(2,n)-1):
    pomoc = zamiana(i)
    rozmiar = 0
    wartosc = 0
    for j in range(n):
        if pomoc[j]==1:
            rozmiar+=lista[j][0]
            wartosc+=lista[j][1]
        if rozmiar<=b:
            if wartosc>max_w:
                max_w=wartosc
                max_r=rozmiar
            wynik = pomoc.copy()

    return wynik, max_w, max_r

def zamiana(x):
    lista_sil = []
    for i in range(n):
        if x>0:
            lista_sil.append(x%2)
            x//=2
        else:
            lista_sil.append(0)
            x//=2
    return lista_sil

def choose():
    print("1. programowanie dynamiczne\n2. algorytm zachlany\n3. algorytm
siłowy")
    ch=input()
    start = timer()
    if ch=='1':
        AD()
    elif ch=='2':
        AZ()
    elif ch=='3':
        elem, wartosc, rozmiar = silowy()
        odp=[]
        for i in range(n):
            if elem[i]==1:
                odp.append(i+1)
        print("Elementy w plecaku: ", odp, "\nRozmiar elementów: ", rozmiar,
"\nWartość elementów: ", wartosc)
    else:
        return
    stop = timer()
    print("%s ms"%((stop-start)*1000))

print("1. Wpisać\n2. File\n0. Exit")
a=input()
while a!= '0':
    if a == '1':
        n,b,lista = wpis()
        choose()
    elif a=='2':
        lista = []
        (n,b) = file()
        choose()
    elif a=='3':

```

```
n = int(input())
b = int(input())
r = 0
while r < 5:
    lista = []
    rand()
    choose()
    n+=50
    b+=5
    r+=1
print("1. Wpisac\n2. File\n0. Exit")
a = input()
```