Dumitru Florentin Giuliano

Grupa 233

Knapsack problem

Versiunea pseudopolinomiala

```
def KnapsackPS(sir = [], k = 0):
    n = len(sir)
    sol = [0 for i in range(k+1)]
    mul  = [[] for i in range(k+1)]
    for i in range(1,k+1):
        maxi = 0
        nr = None
        nr1 = None
        for j in range(n):
            if i - sir[j] >= 0:
                if maxi < sol[i-sir[j]] + sir[j] and not j in mul[i-sir[j]]:
                    maxi = sol[i-sir[j]] + sir[j]
                    nr = j
                    nr1 = i-sir[j]
        if maxi < sol[i-1]:
            sol[i] = sol[i-1]
            mul[i] = mul[i-1].copy()
            continue
        if nr != None:
            mul[i] = mul[nr1].copy()
            mul[i].append(nr)
            sol[i] = maxi
    print(f"Knapsack Pseudopolinomial: {sol[k]}\n")
```

Versiunea algoritmului aproximativ

```python
def KnapsackAprox():
    suma = 0
    u = None
    n = int(input())
    k = int(input())
    n = max(n,0) # daca n < 0, folosim doar 3 variabile, si pentru  a avea spatiu
    while n:
        n -= 1
        i = int(input())
        if u == None:
            u = i
            if suma + i <= k:
                suma += i
        else:
            if suma + i <= k:
                suma += i
                u = i
            else:
                if suma - u +i > suma and suma - u + i <= k:
                    suma =suma - u + i
                    u = i
    print(f"Knapsack Aproximativ: {suma}\n")
```

Demonstratie:

Presuspunem algoritmul Optim = OPT

Presupunem algoritmul nostru ca fiind SOL

cu sihuranta SOL <= OPT

Vrem sa verificam ca 1/2 * OPT <= SOL

preusupenm ca 1/2*OPT > SOL

Vom arata contrariul

Exemplul 1:

in cazul in care numere sunt in ordine descrescatoare => cu siguranta rezultatul este mai mare sau egal decat jumatete decat optim, deoarece se iau de la valorile cele mai mari cat se poate

in cazul in care sunt in ordine crescatoare => cu siguranta va fi cel putin jumatate deoarece se iau valori cat de mult se poate

Dar cel mai bine se demonstreaza pintru - un contra exemplu:

sir = [1,2,10,20]

k = 12

in cazul nostru rezultatul este 11 care este mai mare decat 6

Acest lucru se intampla, deaorece mereu se incearca sa se actualizele suma scazandu -se ultimul numar si sa se adauge cu unul mai mare, daca se poate, acest lucru cand nu se poate adauga un numar nou la suma

Cele 3 variabile de tip int sunt k, suma, si u care reprezinta ultima valoare adaugata