TUTORIAT 8

Linkuri utile:

- https://www.python.org/
- https://docs.python.org/3/
- https://www.codecademy.com/learn/learn-python-3
- https://www.learnpython.org/
- https://stanfordpython.com/#/

Comentariu multiplu PyCharm: CTRL + /

Ce conține tutoriatul ?

I. Metoda Greedy
II. Metoda Divide et Impera
III. Probleme

I.Metoda Greedy

Problemă (Stații)

Patronul unei companii private de transport în comun a primit de la primăria orașului aprobarea de a putea folosi o parte dintre stațiile Regiei Locale de Transport în comun.Stațiile disponibile sunt plasate de-a lungul arterei principale a orașului.

El hotărăște să introducă o cursă rapidă care să străbată orașul, de la un capăt la altul, pe artera principală. Pentru început se ocupă de stațiile situate de aceeași parte a drumului.Patronul are o dilemă: dacă opririle vor fi prea dese, atunci străbaterea orașului va dura prea mult și va plictisi călătorii, iar dacă stațiile sunt prea rare, călătorii vor fi prea puțini. De aceea, criteriile după care patronul stabilește stațiile în care va opri cursa rapidă sunt:

- între doua stații alăturate să fie cel puțin x metrii;
- numărul total de stații să fie maxim.

Ajutați patronul să aleagă stațiile!

Stațiile situate pe aceeași parte a arterei principale sunt numerotate, în ordine, de la un capăt la altul cu 1, 2,..., n.

Exemplu:

10 60 # numărul de stații de pe artera principală, x-numărul de metrii dintre două stații

100 50 25 25 50 10 10 80 20 # distanța dintre stația i și stația i+1, i=1,..,n-1

Rezultatul:

5 #numărul maxim de stații alese de patron

1 2 5 8 9 # numărul de ordine al stațiilor

Se dă: o mulțime de candidați (stațiile de pe artera principală a orașului).

Se cere: o submulțime $S \subseteq C$ (submulțimea stațiilor alese de patron) care:

- îndeplinește condiții interne (distanța minimă dintre două stații este x metrii)
- este optimă (numărul de stații este maxim)

Modalitate de rezolvare

Fie mulțimea ordonată a stațiilor disponibile C = {1, 2,...,n}. Corespunzător acesteia cunoaștem cele n-1 distanțe dintre oricare două stații alăturate: $a_1, a_2, \ldots, a_{n-1}$.

Trebuie să determinăm o submulțime $S \subseteq C$ cu număr maxim de elemente(stații), $S=\{i_1, i_2, \ldots, i_k\}$ în care să se păstreze ordinea din C și să avem îndeplinită condiția $ai_{j+1}-ai_j \leq x$, pentru orice j=1,2,...,k-1.

Se începe cu selectarea stației 1. Se parcurge șirul stațiilor și se caută prima stație care se află la o distanță suficient de îndepărtată față de 1. Se adaugă la soluție stația găsită.

Pseudocod funcție de selecție Greedy:

```
greedy(C, S):
        S = {1}
        dist_ultim = 0
        pentru j = 2, n execută:
                dacă a_{j-1} + dist_ultim ≥ x atunci
                        S = S U {j}
                        dist_ultim = 0
                altfel
                        dist_ultim = dist_ultim + a_{j-1}
                sfârșit dacă
        sfârșit pentru
```

Demonstrație corectitudine

PASUL 1: Demonstrăm că există întotdeauna soluție optimă care conține stația având numărul 1.

Notație: dist(i, j) – distanța dintre stațiile i și j

Fie $S \subseteq C$ o soluție optimă a problemei, unde $S = \{j_1, j_2, \ldots, j_k\}$ nu conține elementul 1(stația 1). Atunci $dist(j_1, j_2) \geq x$. Dar cum stația 1 este situată înaintea stației j_1, avem relația: $dist(1, j_2) = dist(1, j_1) + dist(j_1 + j_2)$, deci $dist(1, j_2) \geq x$. Prin urmare și mulțimea S\` = $\{1, j_2, j_3, \ldots, j_k\}$ îndeplinește condițiile problemei și are numărul de elemente egal cu cel al mulțimii S.

În concluzie, S\` este o soluție optimă.

Am demonstrat că există soluție optimă a problemei care conține stația cu numărul 1. În continuare vor putea fi selectate în S doar acele stații care, față de stația 1, se situează la o distanță cel puțin egală cu x.

PASUL 2: Demonstrăm că metoda de selecție este corectă(conduce la soluția optimă).

Fie S o soluție optimă a problemei. Arătăm că S\`\` = S – {1} construită cu funcția de selecție greedy este o soluție optimă pentru C\` = C – {j | dist(1, j) < x }.

Presupunem că S\`\` nu este o soluție optimă. Deci, există S\`\`\`, astfel încât | S\`\`\`| > | S\`\`|. Dar atunci mulțimea are cardinalul | {1} U S\`\`\`| > | S|, ceea ce este o contradicție cu presupunerea că S este soluție optimă.

Așadar, mulțimea S\`\`, construită cu funcția de selecție greedy, are număr maxim de elemente.Prin urmare și mulțimea S = {1} U S\`\` va avea număr maxim de elemente, deci va fi o soluție a problemei date.

II.Metoda Divide et Impera

Metoda constă în două etape:

1.*DIVIDE* – problema dată este împărțită în două sau mai multe subprobleme de același tip, dar de dimensiuni mai mici; subproblemele se rezolvă direct, dacă dimensiunea lor permite aceasta(elementare) sau, fiind de același tip, se rezolvă în mod recursiv, prin același procedeu.

2.*IMPERA* – se combină soluțiile subproblemelor pentru a obține soluția problemei inițiale.

Descrierea generală a funcției Divide et Impera

```
divide(p,u):
    dacă u – p < e atunci
        r = prelucrare(p,u) #problema se rezolvă direct
    altfel
        m = pozitie(p,u) #de obicei mijlocul
        r1 = divide(p,m)
        r2 = divide(m+1, u)
```

```
            solutia = combina(r1, r2)
        return solutia
```

Apel funcție: divide(1, n) # n – dimensiunea inițială a problemei

Exemplul 1 (Cel mai mare divizor comun)

Se dau n valori naturale. Determinați cel mai mare divizor comun al celor n valori prin metoda Divide et Impera.

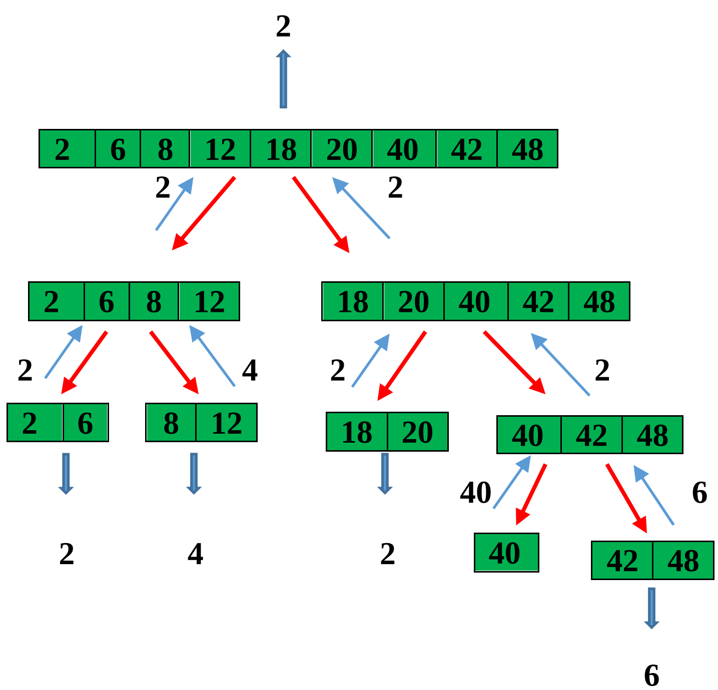
Se cunoaște a fi adevărată relația:

$\text{cmmdc}(a_0,a_1,\ldots,a_{n-1}) = \text{cmmdc}(\text{cmmdc}(a_0,a_1,\ldots,a_m), \text{cmmdc}(a_{m+1},a_{m+2},\ldots,a_{n-1}))$

Exemplu

9

2 6 8 12 18 20 40 42 48 => cmmdc(2, 6, 8, 12, 18, 20, 40, 42, 48) = 2



Exemplul 2 (Sortare prin interclasare)

Se dau n numere. Să se ordoneze crescător numerele.

Metoda de rezolvare

PAS 1: se împarte vectorul în doi subvectori

PAS 2: se ordonează crescător fiecare subvector

PAS 3: se combină rezultatele prin interclasare

Exemplu

PAS 1:

PAS 2:

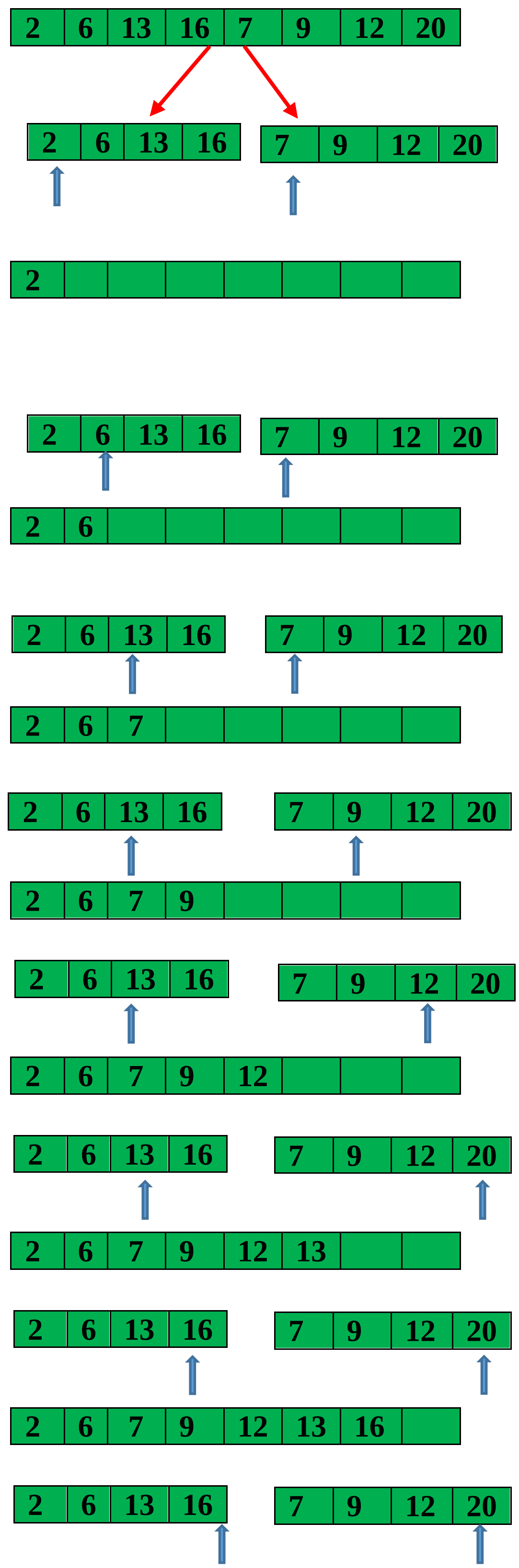
PAS 3:

PAS 4:

PAS 5:

PAS 6:

PAS 7:



2	6	7	9	12	13	16	20

Complexitate (O(n log(n))

$T(n) = 2T(n/2) + cn,\ n > 1,\ n = 2^k$

$T(n) = T(2^k) = 2T(2^{k-1}) + c*2^k = 2\,[2T(2^{k-2}) + c*2^{k-1}] + c*2^k = 2^2T(2^{k-2}) + 2*c*2^k =$

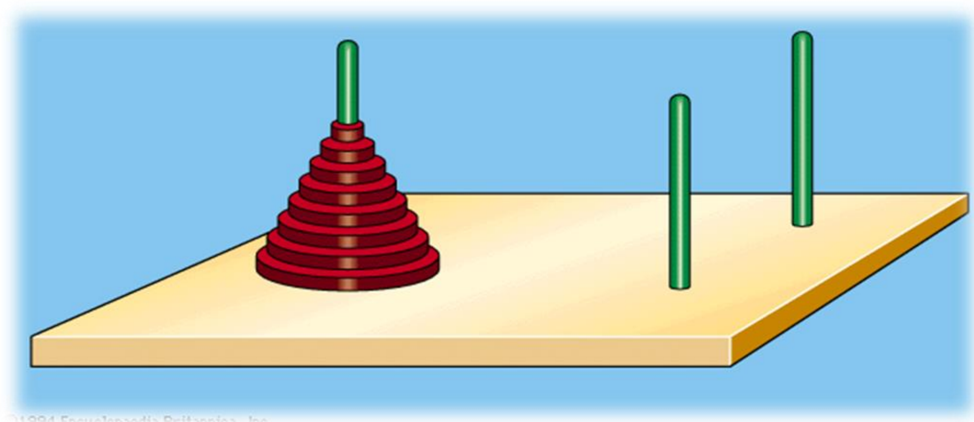
$\ldots\ldots = 2^iT(2^{k-i}) + i*c*2^k = 2^kT(1) + k*c*2^k = nc + c * n *\log_2 n$

- ⇨ O(n log(n))

Exemplul 3 (Problema Turnurilor din Hanoi)

Avem la dispoziție trei tije 1, 2, 3 și n discuri de diametre diferite. Inițial, toate discurile sunt plasate pe tija 1 în ordinea descrescătoare a diametrelor, considerând sensul de la bază spre vârf. Problema constă în a muta discurile de pe tija 1 pe tija 2, folosind ca tijă de manevră tija 3, respectând următoarele reguli:

- la fiecare pas se mută un singur disc;
- orice disc poate fi așezat doar peste un disc cu diametru mai mare



Metoda de rezolvare

Notație: i -> j mutare , i, j apartine {1, 2, 3} i ≠ j – se mută un disc de pe tija i pe tija j.

Dacă n = 1, atunci avem de mutat un singur disc, problema se rezolvă direct(caz elementar).

Dacă n > 1, pentru a muta discul de diametru maxim de pe tija i pe tija j, trebuie să îl „eliberăm”, adică să plasăm cele n-1 discuri de deasupra sa pe tija de manevră. Cele 3 tije fiind numerotate distinct cu valori din mulțimea 1, 2, 3 , suma numerelor tijelor este 6, deci tija de manevră este numerotată 6 – i– j. Discul de diametru maxim fiind eliberat, poate fi mutat pe tija j, după care cele n-1 discuri de pe tija 6 – i – j pe tija j, folosind de data aceasta tija i, care este goală, ca tijă de manevră.

Funcția H(x,i,j) șirul de mutări necesare pentru x discuri de pe tija i pe tija j:

$$H(x,i,j) = \begin{cases} i \to j, & x = 1 \\ H(x-1,i,6-i-j)\, i \to j\, H(x-1,6-i-j,j) \end{cases}$$

III.Probleme

1. Fie o tablă cu pătrăţele de dimensiune 2^N * 2^N (cu N citit de la tastatură). Pe această tablă există o gaură la o poziţie dată prin linia şi coloana sa (lg, cg) (liniile şi coloanele se consideră numerotate de la 1). Pentru acoperirea acestei table avem la dispoziție piese de forma unui L.

Aceste piese pot fi rotite cu 90, 180 sau 270 de grade. Să se afișeze o acoperire completă a tablei (cu excepţia găurii). Piesele vor fi reprezentate prin numere de la 1 la K, iar gaura prin 0 (cele 3 căsuţe ocupate de a i-a piesă pusă vor primi valoarea i). (problemă preluată laborator grupa 144, an universitar 2020-2021)

Exemplu(soluția nu este unică)

Date de intrare	Date de ieșire
2	1 1 2 2
3 1	1 3 3 2
	0 4 3 5
	4 4 5 5