**Explicații algoritmi și pseudo cod**

**Problema rucsacului**

**Pentru Algoritm 1**

1. Pas 1
2. Pas 2.

**Parametrul algoritmului este k.**

func generare\_k\_solutii(k: int):  
 solutii = []  
 <cod>  
 returneaza solutii

Se returnează soluția cu valoarea cea mai mare.

func alg1(obiecte: list, solutii: list, capacitate\_maxima: int):  
 best = []  
 <mai mult cod>  
 returneaza best

**Pentru Algoritm 2**

1. Pas 1
2. Pas 2
3. Pas 3
4. Returnează cel mai bun c obținut.

**Parametri algoritmului sunt k si p.**

func alg2(k: int, p: int):  
 <cod>  
 returneaza cea\_mai\_buna\_solutie

**Tabele de date**

**Pentru Alg1**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Instanța problemei | k | Valoare medie | Valoarea cea mai bună | Număr execuții | Timpul mediu de execuție |
| Inst 1 |  |  |  | 10 |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
| Inst 2 |  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
| Inst 3 |  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

**Pentru Alg2**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Instanța problemei | K | p | Valoare medie | Valoarea cea mai bună | Număr execuții | Timpul mediu de execuție |
| Inst 1 |  |  |  |  | 10 |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
| Inst 2 |  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
| Inst 3 |  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

**Observații**

*Următoarele observații sunt date în funcție de datele din tabelele de mai sus.*

**Concluzie**

**Datele obținute după rulări**