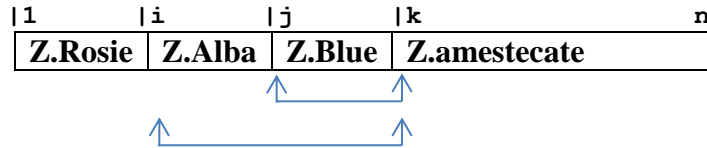


Se dă o secvență de bile de 3 culori roșu, alb, blue amestecate . Să se pună în grupuri compacte de culori parcurgând o singură dată secvența de la stânga la dreapta..



```
lung(Z.Rosie) = i-1
```

lung(Z.Alba) = j-i

lung(Z.Blue) = k-j

lung(Z.amestecate)=n-k+1

```

Algorithm DutchFlag(n,B[])           //B[1],B[2],..., B[n]
    i=j=k=1                          //zonele de culori sunt vide la inceput
                                    //Z.amestecate este de lungime n

    CatTimp (k<=n) executa
        Daca B[k]="Blue"
            atunci k ← k+1           //Z.Blue s-a marit cu o bila
            altfel
                Daca B[k]="Alb"
                    atunci B[k] ↔ B[j] //se interschimba bila alba cu prima blue
                    j ← j+1           //Z.Alba s-a marit cu o bila
                    k ← k+1           //Z.Blue s-a deplasat la dreapta cu o pozitie
                altfel B[k] ↔ B[i] //se interschimba rosie cu prima alba
                    i ← i+1           //Z.Rosie s-a marit cu o bila
                    B[k] ↔ B[j] //se interschimba alba cu prima blue
                    j ← j+1           //Z.Alba s-a deplasat la dreapta cu o pozitie
                    k ← k+1           //Z.Blue s-a deplasat la dreapta cu o pozitie

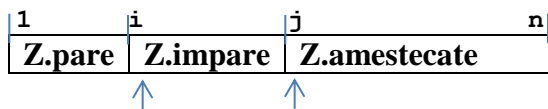
        SfDaca
    SfDaca
SfCatTimp
SfAlgorithm

```

Problemă de Dijkstra (dedusă din problema Dutch Flag)

Se dă o secvență S de numere naturale, să se pună numerele pare în partea de început a secvenței.

Exemplu: 44,45,46,47,48,49,50 => 44,46,48,50,45,47,49.



lung(Z.pare)=i-1

lung(Z.impair)=j-i

lung(Z.amestecate)=n-j+1

```
Algoritm PareImpair(n, S[])
    i=j=1 //zonele pare si impar sunt vide
    CatTimp(j<=n) executa
        Daca S[j]=impar
            atunci
                j ← j+1 //creste Z.impair
            atunci
                S[i] ↔ S[j] //se interchimba S[i] cu S[j]
                i ← i+1 //Z.pare creste
                j ← j+1 //Z.impair s-a deplasat la dreapta
    SfDaca
    SfCatTimp
    SfAlgoritm
```

41. Să se determine sume de numere consecutive care dau exact **n**, **n** dat.

Exemplu n=15 7+8, 4+5+6, 1+2+3+4+5

Algoritm SumaCons (n) este:

```
Pt i ← 1, [n/2] executa
    Suma ← i
    j ← i
    CatTimp (Suma<n) executa
        Suma ← Suma+j
        j ← j+1
    SfCatTimp
    Daca (Suma=n)
        Atunci pt k←i, j-1 executa
            Afis(k, "+")
            SfPt
    SfDaca
    SfPt
SfAlgoritm
```