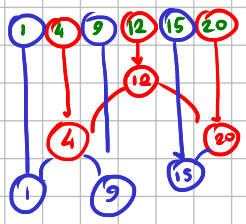
Esercizio 1 (Per fare l'albero ci vuole...). *Sia* BinSearchTree *una classe descrivente un nodo di un albero binario di ricerca (ABR), dotata di un campo* key *e due puntatori* sx, dx *al figlio sinistro e destro del nodo.*

Progettare un algoritmo di complessità $O(n)$ che dato in input un array A di n interi distinti ordinati in modo crescente restituisca un ABR di altezza minima contenente gli interi di A.




```
ric(A:array){
    n=A.length()
    if(n==1){ return A[0] }
    i=⌈n/2⌉
    u=crea_nodo(A[i-1])
    u.sx=ric(A[0:i+1])
    u.dx=ric(A[i:n])
    return u
}
```

Esercizio 2 (MST senza arco specifico). *Sia G un grafo non diretto, pesato e connesso. Progettare un algoritmo che dato in input il grafo G e un arco $(x, y) \in E(G)$ restituisca True se esiste un albero $T \subseteq G$ tale che T sia un MST di G e che $(x, y) \notin E(T)$ oppure False se tale T non esiste.*

```
Es2(G:grafo, (x,y):arco){
    G'=G\(x,y)
    T=Kruskal(G)
    T'=Kruskal(G')
    if(ω(T)==ω(T')){ return True }
    return False
}
```

Esercizio 3 (5-copertura di costo minimo). *Data una sequenza X di n interi positivi, definiamo come 5-copertura di X una sotto-sequenza A di suoi elementi se, presi 5 elementi consecutivi qualsiasi della sequenza di X, almeno uno di questi è in A.*

Dare lo pseudocodice di un algoritmo di complessità $O(n)$ che data una sequenza X in input restituisca una 5-copertura di costo minimo e il suo costo. Ad esempio, per $X = [2, 4, 1, 2, 6, 4, 8, 3, 5, 1]$ una 5-copertura di costo minimo è $[\_, \_, 1, \_, \_, \_, \_, 3, \_, \_]$, avente costo pari a 4

```
cop(X:array){
    A[n]=[_,_,....,_]
    i=0
    while(i<n){
        in:int
        min=∞
        For(J=0,1...,4){
            if(i+J≥n){return A}
            if(X[i+J]<min){
                min=X[i+J]
                in=i+J
            }
        }
        A[i+J]=X[i+J]
        i=i+J
    }
    return A
}
```