```
def valInf( T,v , bg =None ,bd =None ):
    if bg ==None
        L --
    if bg==bd return T[bg]
    if bd == bg+1
        if T[bd] < v : return T[bd]
        else return T[bg]
    p = (bg+bd)/2
    if v>m : return valInf(T,v ,p ,bd)
        else return valInf (T,v,bg,p-1)
```

~ correction exam recherche dicho ~

Analyse amortie de complexité

Idée : → on analyse une séquence d'instructions (en général une SDD avec plusieurs opérations)

→ on pourra tenir compte du fait qu'une opération coûteuse est rare et la séquence totale a une bonne complexité

▷ ≠ complexité moyenne

▷ complexité dans le pire cas

∃ 3 Méthodes :

- → évaluer la complexité $T(n)$ de la séquence complète
  → on dit que chaque op a une complexité amortie $\simeq$? à $\frac{T}{n}$

- Attribuer des "charges" pr chaque opération
  ↳ coût amorti de l'opération.

  → si charge > coût réel de l'op, la ≠ est un crédit attribué à certains objets de la SDD

  → si charge < coût réel, utilise des crédits déjà stockés dans l'opération

- on déf une fonction potentiel (Φ) à la SDD $\Phi(D)$ correspond à une quantité d'énergie stockée dns la struct.

  Une opération peut conduire à :

  - ↗ le potentiel : emmagasine les réserves
  - ↘ le potentiel : dépense les réserves

coût réel, potentiel après i-ème op

coût i-ème contribution de f par : $\hat{c}_i = c_i + \Phi(D_i) - \Phi(D_{i-1})$ ← potentiel avant op ex Pile opération :

- pop
- push
- multipop :

```
while P ≠ ∅ ∧ k > 0 :
    Pop (P)
    k--
```

→ complexité ?

- pop, push = 1

multipop = min(k, n) → ~~tpf~~ --

→ ~~O(n)~~

→ amorti $\frac{O(n)}{n} = O(1)$ car on analyse n opérations et non les op 1 par 1

Rq : tab dynamique utilise car (mult tab ×2)
→ en moyenne ✓

Méthodes

n cas opération pile avec pile → de manière fixe

1)
push 2
pop 0
multipop 0
→ O(n)

2)
push $\hat{c}_i = 1 + \underbrace{\Phi(P_i) - \Phi(P_{i-1})}_{=1} = 2$

pop $\hat{c}_i = 1 + \underbrace{\Phi(P_i) - \Phi(P_i - 1)}_{=-1} = 0$

multipop $\hat{c}_i = k' + \underbrace{\Phi(P_i) - \Phi(P_i - 1)}_{=-k'} = 0$

$k' = \min(k, |p|)$