TD 3

Alain Cournier

Cours de MFADE Licence 3 Informatique UPJV

Recherche du minimum : Spécification

T Est un tableau indicé de 1 à N (énoncé)

Précondition : T Non vide c'est-à-dire N > 1

Spécif : IndiceValeurMinimale(T) = u si et seulement si :

1. u est un entier naturel compris entre 1 et N (au sens large)
2. Pour tout entier naturel j compris entre 1 et N (au sens large) T[j] ≥ T[u]

Recherche du minimum V. récursive : Idée 1

Plaçons nous à une étape $i < N$ du programme lors de cette étape nous allons chercher l'indice de la valeur minimale dans la partie du tableau T comprise entre les indices i et N.

Lorsque $i < N$:

1. On cherche (récursivement) l'indice j de la valeur minimale dans la partie du tableau T comprise entre les indices $i+1$ et N.
2. On compare $T[i]$ et $T[j]$:
   a) $T[j] < T[i]$: le résultat sera j
   b) $T[j] \geq T[i]$: le résultat sera i

Recherche du minimum V. récursive : Idée 1

Peut-on commencer ? Oui nous allons chercher l’indice de la valeur minimale dans la partie du tableau T comprise entre les indices 1 et N.

Quand s’arrêter ? On s’arrête dès que i = N et dans ce cas on renvoie la valeur N.

Recherche du minimum V. réc : Fonc 1

Fonction IndiceValeurMinimale1

Donnée : T : Tableau d'entier indicé de 1 à N ;

Résultat : un entier

DébutCode

```
    Renvoyer (IVM1Rec(T,1))
```

FinCode

Fonction IVM1

Donnée :

T : Tableau d'entier indicé de 1 à N ; \\ T non vide

i : Un indice de T

Résultat : un entier

Variable j un entier

DébutCode

```
Si i=N alors renvoyer (N)
Sinon
    j ← IVM1(T, i+1)
    Si T[j] < T[i] alors renvoyer (j) sinon renvoyer (i) FinSi
FinSi
```

FinCode

Recherche du minimum V. récursive : Idée 2

Plaçons nous à une étape du programme lors de cette étape nous allons chercher l'indice de la valeur minimale dans la partie du tableau T comprise entre les indices d et f (compris).

Si $d \neq f$ alors on cherche un indice u tel que $d \leq u < f$:

1. On cherche (récursivement) l'indice i de la valeur minimale dans la partie du tableau T comprise entre les indices d et u.
2. On cherche (récursivement) l'indice j de la valeur minimale dans la partie du tableau T comprise entre les indices u+1 et f
3. On compare T[i] et T[j] :
   a) $T[j] < T[i]$: le résultat sera j
   b) $T[j] \geq T[i]$: le résultat sera i

Recherche du minimum V. récursive : Idée 2

Peut-on commencer ? Oui nous allons chercher l'indice de la valeur minimale dans la partie du tableau T comprise entre les indices 1 et N.

Quand s'arrêter ? On s'arrête dès que d = f et dans ce cas on renvoie la valeur de la variable d.

Recherche du minimum V. réc : Fonc 2

Fonction IndiceValeurMinimale2

Donnée : T : Tableau d'entier indicé de 1 à N ;

Résultat : un entier

DébutCode

```
    Renvoyer (IVM2Rec(T,1,N))
```

FinCode

Fonction IVM2Rec

Données :

T : Tableau d'entier indicé de 1 à N ; \\ T non vide

d,f : deux indices de T \\(préconditions : d ≤ f)

Résultat : un entier Variables : u, i, j entiers

DébutCode

```
    Si d=f alors renvoyer (d)
    Sinon u ← (d+f) div 2
        i ← IVM2Rec(T, d, u) ; j ← IVM2Rec(T, u+1, f)
        Si T[j] < T[i] alors renvoyer (j) sinon renvoyer (i) FinSi
    FinSi
```

FinCode

Recherche du minimum V. réc2 :

Remarque 1 : $d < f$ implique $d \leq (d+f)$ div $2 < f$.

NB : Dans la fonction si $d<f$ alors $[d,u] \cup [u+1,f] = [d,f]$

Conséquence 1 : si d et f sont des indices de T tels que $d < f$ alors :
$(d+f)$ div 2 est un indice de T

Conséquence 2 : si d et f sont des indices de T tels que $d < f$ alors :
$((d+f)$ div $2) + 1$ est un indice de T

Recherche du minimum V. réc2 : Arrêt

PredArrêtIVM2(T, d, f) : Soit T un tableau non vide, soient d et f deux indices de T tels que d ≤ f, alors IVM2Rec(T, d, f) s'arrête en un temps fini.

La preuve se fait par récurrence sur la valeur de k=f-d. (f=d+k)

PredArrêtIVM2(T, d, d+k) : Soit T un tableau non vide, soient d et d+k deux indices de T tels que d ≤ d+k, alors IVM2Rec(T, d, d+k) s'arrête en un temps fini.

Recherche du minimum V. réc2 : Arrêt

Pour k = 0, d=f IVM2Rec effectue le test d=f puis renvoie d donc le prédicat PredArrêtIVM2(T, d, d+0) est vrai

Hypothèse de récurrence : Il existe un entier naturel k tel que pour tout entier t, $0 \leq t \leq k$, PredArrêtIVM2(T, d, d+t) est vrai.

Démontrons que PredArrêtIVM2(T,d,d+k+1) est vrai.

Remarque 2 : Si k est un entier naturel $d < d + k + 1$.

Recherche du minimum V. réc2 : Arrêt

Lançons l'exécution de IVM2(T,d,d+k+1)

Puisque $d < d + k + 1$, on passe par la partie sinon

- $u = (d+d+k+1)$ div $2 = (2d+k+1)$ div 2 compte tenu de la Remarque 1, nous pouvons affirmer que $d \leq u < d + k + 1$
- On effectue ensuite l'instruction i $\leftarrow$ IVM2Rec(T, d, u). Or puisque $u < d + k + 1$ nous pouvons affirmer que $u \leq d + k$ donc par hypothèse de récurrence le prédicat PredArrêtIVM2(T, d, u) est vrai donc la fonction IVM2Rec(T, d, u) s'arrête et renvoie un résultat.

Suite page suivante

Recherche du minimum V. réc2 : Arrêt

La suite

- On effectue ensuite l'instruction j ← IVM2Rec(T, u+1, d+k+1). Or puisque d ≤ u nous pouvons affirmer que d < u+1, donc d+k+1 ≤ (u+1)+k donc par hypothèse de récurrence le prédicat PredArrêtIVM2(T, u+1, d+k+1) est vrai donc la fonction IVM2Rec(T, u+1, d+k+1) s'arrête et renvoie un résultat.
- On effectue la comparaison entre T[i] et T[j] ce test se finira si et seulement si les indices i et j renvoyés sont valides (à démontrer avec la preuve du bon résultat).
- En fonction du résultat du test IVM2Rec renvoie i ou j et se termine.
- Donc sous réserve que les résultats renvoyés sont bien des indices le prédicat PredArrêtIVM2(T, d, d+k+1) est vrai

Recherche du minimum V. réc2 : Bon résultat

PredBonResIVM2(T, d, f) : Soit T un tableau non vide, soient d et f deux indices de T tels que alors d ≤ f, IVM2Rec(T, d, f) = ind si et seulement si les deux conditions suivantes sont vérifiées

1. Pour tout entier w, compris entre d et f (au sens large), T[ind] ≤ T[w].
2. d ≤ ind ≤ f et ind est un indice de T

La preuve se fait par récurrence sur la valeur de k=f-d. (f=d+k)

Recherche du minimum V. réc2 : Bon résultat

La preuve se fait par récurrence sur la valeur de k=f-d. (f=d+k)

PredBonResIVM2(T, d, d+k) : Soit T un tableau non vide, soient d et d+k deux indices de T tels que $d \le d+k$, alors IVM2Rec(T, d, d+k) = ind si et seulement si les deux conditions suivantes sont vérifiées

1. Pour tout entier w, compris entre d et d+k (au sens large), $T[ind] \le T[w]$.
2. $d \le ind \le d+k$ et ind est un indice de T

Recherche du minimum V. réc2 : Bon résultat

Pour $k = 0$, $d=f$ IVM2Rec effectue le test $d=f$ puis renvoie $ind = d$ donc :

1. Pour tout entier w, compris entre d et $d+0$ (au sens large), $T[ind] \leq T[w]$.
2. $d \leq ind \leq d+0$ et ind est un indice de T

Le prédicat PredBonResIVM2(T, d, d+0) est donc vérifié.

Hypothèse de récurrence : Il existe un entier naturel k tel que pour tout entier t, $0 \leq t \leq k$, PredBonResIVM2(T, d, d+t) est vrai.

Démontrons que PredBonResIVM2(T,d,d+k+1) est vrai.

Recherche du minimum V. réc2 : Bon résultat

Lançons l'exécution de IVM2Rec(T,d,d+k+1)

Puisque $d < d + k + 1$, on passe par la partie sinon

- $u = (d+d+k+1)$ div $2 = (2d+k+1)$ div 2 compte tenu de la Remarque 1, nous pouvons affirmer que $d \leq u < d + k + 1$
- On effectue ensuite l'instruction $i \leftarrow$ IVM2Rec(T, d, u). Or puisque $u < d + k + 1$ nous pouvons affirmer que $u \leq d + k$ donc par hypothèse de récurrence le prédicat PredBonResIVM2(T, d, u) est vrai en conséquence i est un indice de T compris entre d et u et i est l'indice de la valeur minimale du tableau entre les indices d et u (compris).

Suite page suivante

Recherche du minimum V. réc2 : Bon résultat

La suite

On effectue ensuite l'instruction j ← IVM2Rec(T, u+1, d+k+1). Or puisque d ≤ u nous pouvons affirmer que d < u+1, donc d+k+1 ≤ (u+1)+k donc par hypothèse de récurrence le prédicat PredBonResIVM2(T, u+1, d+k+1) est vrai en conséquence j est un indice de T compris entre u+1 et d+k+1 et j est l'indice de la valeur minimale du tableau entre les indices u+1 et d+k+1 (compris).

Suite page suivante

Recherche du minimum V. réc2 : Bon résultat

La suite

On effectue le test T[j] < T[i] nous devons distinguer deux cas :

Cas 1 : T[j] < T[i] répond vrai. La fonction renvoie j. Puisque T[i] était la valeur minimale de T entre les indices d et u (compris) et que T[j] < T[i], T[j] est strictement inférieur aux valeurs contenues dans le tableau entre les indices d et u. De plus j est l’indice de la valeur minimale du tableau entre les indices u+1 et d+k+1 (compris). Donc j est l’indice de la valeur minimale du tableau entre les indices d et d+k+1 (compris). Enfin j compris entre u+1 et d+k+1 (au sens large) implique que j est compris entre d et d+k+1 (au sens large).

Le Predicat PredBonResIVM2(T,d,d+k+1) est vrai dans ce cas.

Suite page suivante

Recherche du minimum V. réc2 : Bon résultat

La suite

- Cas 2 : T[j] < T[i] répond faux. La fonction renvoie i. Puisque T[j] était la valeur minimale de T entre les indices u+1 et u+k+1 (compris) et que T[i] ≤ T[j], T[i] est strictement inférieur aux valeurs contenues dans le tableau entre les indices u+1 et d+k+1. De plus i est l'indice de la valeur minimale du tableau entre les indices d et u (compris). Donc i est l'indice de la valeur minimale du tableau entre les indices d et d+k+1 (compris). Enfin i compris entre d et u (au sens large) implique que i est compris entre d et d+k+1 (au sens large).
- Le Predicat PredBonResIVM2(T,d,d+k+1) est vrai dans ce cas.

Dans tous les cas Le Predicat PredBonResIVM2(T,d,d+k+1) est vrai.

Recherche du minimum V. réc2 : Complexité

Posons n = f-d. et T(n) la fonction qui permet de calculer la complexité de notre fonction INV2Rec (T,d,d+n)

$T(0) = 4$

$T(1) = 3 + 2T(0) + 15 = 26$

Si $n > 1$

$T(n) = 2T(n/2) + 18 = 4T(n/4) + 18\ (1 + 2) = 8T(n/8) + 18\ (1+2+4)$

$T(n) = 2^i T(n/2^i) + 18\ (2^i - 1) = 2^{\log n} T(n/2^{\log n}) + 18\ (2^{\log n} - 1) = n\ T(1) + 18\ n$

$T(n) = 44\ n$ Donc $T(n) \in \Theta(n)$.