

INFO0947: Projet 1 -> Construction de Programme

Groupe 07: Julien PEFFER, Alexandre BOURGUIGNON

Table des matières

1	Introduction	3
2	Formalisation du Problème	3
2.0.1	Prédicat A	3
2.0.2	Prédicat B	3
2.0.3	Prédicat C	3
2.0.4	Prédicat D	3
3	Définition et Analyse du Problème	4
3.1	Input	4
3.2	Output	4
3.3	Objets Utilisés	4
3.4	Découpe en sous problèmes	4
3.5	Emboitement	4
4	Specifications	4
5	Invariants	5
5.1	Représentation de la Postcondition	5
5.2	Représentation de l'invariant graphique	5
5.3	Invariant Formel	5
6	Approche Constructive	5
7	Code Complet	9
8	Complexité	10
9	Conclusion	12

1 Introduction

L'objectif de ce projet est de créer une fonction qui prend en entrées un tableau, sa taille, un pointeur vers un entier qui correspond à la position du minimum dans le tableau et un autre pointeur vers un entier qui correspond à la position du maximum dans le tableau. Cette fonction va retourner 3 éléments : la position du minimum et du maximum via un passage par adresse et également la somme entre le minimum et le maximum inclus. Il est important de noter que le minimum n'est pas forcément avant le maximum et également que la fonction ne doit contenir qu'une seule boucle de type "while". Par conséquent, la complexité de la fonction doit être en $O(N)$. Nous allons donc parcourir, point par point, chaque étape au moyen d'une approche constructive.

2 Formalisation du Problème

Voici 3 prédicats dont un prédicat intermédiaire. Rappel : un prédicat est une fonction produisant une valeur booléenne.

2.0.1 Prédicat A

La notation est MAXPOS

La formalisation est $MAXPOS(T, Inf, Sup) \equiv (\exists x = \max_{i \in Inf \dots Sup-1}, (T[i])) \wedge (\exists j, Inf \leq j \leq Sup - 1, T[j] = x) \wedge (\exists maxPos, Inf \leq maxPos \leq Sup - 1, maxPos = j)$

2.0.2 Prédicat B

La notation est MINPOS

La formalisation est $MINPOS(T, Inf, Sup) \equiv (\exists y = \min_{i \in Inf \dots Sup-1}, (T[i])) \wedge (\exists j, Inf \leq j \leq Sup - 1, T[j] = y) \wedge (\exists minPos, inf \leq minPos \leq Sup - 1, minPos = j)$

2.0.3 Prédicat C

La notation est SUM

La formalisation est $SUM(T, N, min_pos, max_pos) \equiv (\forall i, 0 \leq i \leq N - 1, sum = \sum_{i=\min(min_pos, max_pos)}^{\max(min_pos, max_pos)} T[i])$

2.0.4 Prédicat D

La notation est NEWSUM

La formalisation est $NEWSUM(T, N, min_pos, max_pos) \equiv (\forall i, 0 \leq i \leq N - 1, \exists new_sum = \sum_{a=\max(min_pos, max_pos)}^{i-1} T[a])$

3 Définition et Analyse du Problème

3.1 Input

- T , tableau d'entiers
- N , taille du tableau
- $*min_pos$, un pointeur vers la position du minimum
- $*max_pos$, un pointeur vers la position du maximum

3.2 Output

La somme des éléments entre la position minimale et la position maximale (inclus)

3.3 Objets Utilisés

- T , tableau d'entiers ($int * T$)
- N , valeur entière ($int N$)
- sum , valeur entière ($int sum$)
- new_sum , valeur entière ($int new_sum$)

3.4 Découpe en sous problèmes

SP 1 : calcul de MAXPOS($T, 0, i$) et de MINPOS($T, 0, i$)

SP 2 : Mettre à jour l'indice de max_pos et min_pos respectivement par MAXPOS($T, 0, i$) et MINPOS($T, 0, i$)

SP 3 : Calcul de SUM(T, N, min_pos, max_pos)

3.5 Emboitement

SP1 \rightarrow (SP3 \subset SP2)

4 Specifications

- Spécification du SP1 & SP2

```
1  /*
2  /*
3  *@pre: T initialisé  $\wedge 0 \leq inf \leq sup \leq N - 1$ 
4  *@post: N = N0  $\wedge T = T_0 \wedge max\_pos = MAXPOS(T, 0, N)$ 
5  /*
6  int max_pos(int *T, int inf, int sup);
7
8  /*
9  *@pre: T initialisé  $\wedge 0 \leq inf \leq sup \leq N - 1$ 
10 *@post: N = N0  $\wedge T = T_0 \wedge min\_pos = MINPOS(T, 0, N)$ 
11 /*
12 int min_pos(int *T, int inf, int sup);
```

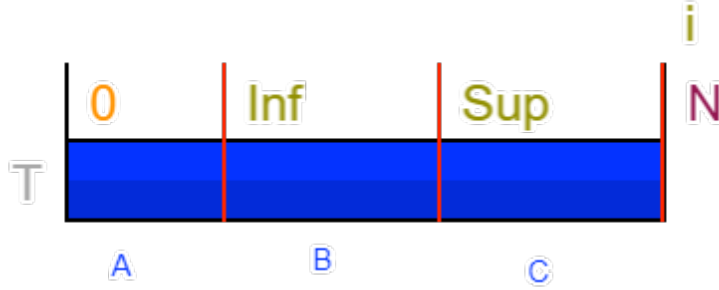
- Spécification du SP3

```
1  /*
2  *@pre: T initialisé  $\wedge N > 0 \wedge min\_pos$  initialisé  $\wedge max\_pos$  initialisé
3  *@post: T = T0  $\wedge N = N_0 \wedge sum = SUM(T, N, min\_pos, max\_pos)$ 
4  /*
```

```
4 int sum(int *T, int N, int *min_pos, int *max_pos);
```

5 Invariants

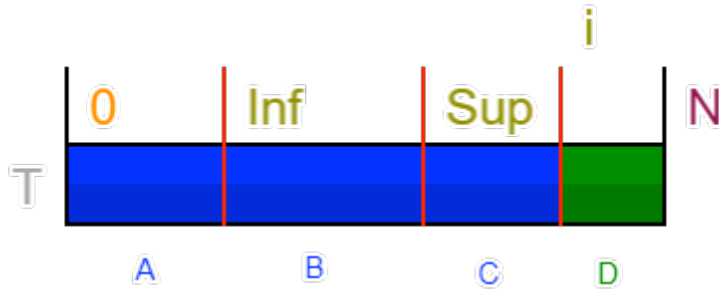
5.1 Représentation de la Postcondition



Avec :

- A -> Déjà parcouru ($\text{sup} \neq \text{MAXPOS}(T, 0, \text{inf})$ et $\text{inf} \neq \text{MINPOS}(T, 0, \text{inf})$)
- B -> $\text{sum} = \text{SUM}(T, \text{Inf}, \text{Sup})$, $\text{inf} = \text{MINPOS}(T, 0, N)$ et $\text{sup} = \text{MAXPOS}(T, 0, N)$
- C -> Déjà parcouru ($\text{sup} \neq \text{MAXPOS}(T, 0, N)$ et $\text{inf} \neq \text{MINPOS}(T, 0, N)$)

5.2 Représentation de l'invariant graphique



Avec :

- A -> Déjà parcouru ($\text{sup} \neq \text{MAXPOS}(T, 0, \text{inf})$ et $\text{inf} \neq \text{MINPOS}(T, 0, \text{inf})$)
- B -> $\text{sum} = \text{SUM}(T, \text{Inf}, \text{Sup})$, $\text{inf} = \text{MINPOS}(T, 0, i)$ et $\text{sup} = \text{MAXPOS}(T, 0, i)$
- C -> $\text{new_sum} = \text{NEWSUM}(T, N, \text{inf}, \text{sup})$
- D -> Reste à parcourir

5.3 Invariant Formel

$T = T_0 \wedge N = N_0 \wedge 0 \leq \text{inf} \leq \text{sup} \leq N \wedge \text{new_sum} = \text{NEWSUM}(T, N, \text{min_pos}, \text{max_pos}) \wedge \text{max_pos} = \text{MAXPOS}(T, 0, i) \wedge \text{min_pos} = \text{MINPOS}(T, 0, i) \wedge \text{sum} = \text{SUM}(T, N, \text{min_pos}, \text{max_pos})$

6 Approche Constructive

- Spécification du problème général

1 /*

```

2  *@pre: T initialisé  $\wedge$   $N > 0 \wedge$  min_pos initialisé  $\wedge$  max_pos initialisé
3  *@post: T = T0  $\wedge$  N = N0   somme = SUM(T, N, min_pos, max_pos)
4  */
5  int somme(T, N, *min_pos, *max_pos);

```

- Approche constructive

```

1  int somme(T, N, *min_pos, *max_pos){
2  assert(T != NULL && N > 0 && min_pos != NULL && max_pos != NULL);
3  //PRE  $\equiv$  T initialisé  $\wedge$   $N > 0 \wedge$  min_pos init  $\wedge$  max_pos init
4
5  int min = T[0];
6  // T = T0  $\wedge$  N = N0  $\wedge$  min = T[0]
7  int max = T[0];
8  // T = T0  $\wedge$  N = N0  $\wedge$  min = T[0]  $\wedge$  max = T[0]
9  int sum = T[0];
10 // T = T0  $\wedge$  N = N0  $\wedge$  min = T[0]  $\wedge$  max = T[0]  $\wedge$  sum = T[0]
11 int new_sum = 0;
12 // T = T0  $\wedge$  N = N0  $\wedge$  min = T[0]  $\wedge$  max = T[0]  $\wedge$  sum = T[0]  $\wedge$  new_sum = 0
13 *min_pos = *max_pos = 0;
14 /*
15 T = T0  $\wedge$  N = N0  $\wedge$  min = T[0]  $\wedge$  max = T[0]  $\wedge$  sum = T[0]
16  $\wedge$  new_sum = 0  $\wedge$  *min_pos = 0  $\wedge$  *max_pos = 0
17 */
18 int i = 0;
19 /*
20 T = T0  $\wedge$  N = N0  $\wedge$  min = T[0]  $\wedge$  max = T[0]  $\wedge$  sum = T[0]
21  $\wedge$  new_sum = 0  $\wedge$  *min_pos = 0  $\wedge$  *max_pos = 0  $\wedge$  i = 0
22  $\Rightarrow$  *min_pos = MINPOS(T, 0, 0) = MINPOS(T, inf, sup)  $\wedge$ 
23 *max_pos = MAXPOS(T, 0, 0) = MAXPOS(T, inf, sup)
24  $\wedge$  sum = SUM(T, 1, 0, 0) = SUM(T, 1, inf, sup)
25  $\wedge$  new_sum = NEWSUM(T, 1, min_pos, max_pos) = NEWSUM(T, 1, inf, sup)
26 */
27 /*
28 INV  $\equiv$  N = N0  $\wedge$  T = T0  $\wedge$   $0 \leq inf \leq sup \leq i \leq N \wedge$  *min_pos = MINPOS(T, 0, i)
29  $\wedge$  *max_pos = MAXPOS(T, 0, i)  $\wedge$  sum = SUM(T, i + 1, *min_pos, *max_pos)
30  $\wedge$  new_sum = NEWSUM(T, i + 1, *min_pos, *max_pos)
31 */
32 while(i < N){
33 /*
34 INV  $\wedge$  B  $\equiv$  N = N0  $\wedge$  T = T0  $\wedge$   $0 \leq inf \leq sup \leq i \leq N - 1 \wedge$  *min_pos = MINPOS(T, 0, i)
35  $\wedge$  *max_pos = MAXPOS(T, 0, i)  $\wedge$  sum = SUM(T, i + 1, *min_pos, *max_pos)
36  $\wedge$  new_sum = NEWSUM(T, i + 1, *min_pos, *max_pos)
37 */
38 new_sum += T[i];
39 /*
40 N = N0  $\wedge$  T = T0  $\wedge$   $0 \leq inf \leq sup \leq i \leq N - 1 \wedge$  *min_pos = MINPOS(T, 0, i)
41  $\wedge$  *max_pos = MAXPOS(T, 0, i)  $\wedge$  sum = SUM(T, i + 1, *min_pos, *max_pos)
42  $\wedge$  new_sum = NEWSUM(T, i + 2, *min_pos, *max_pos)
43 */
44 if(T[i] < min){
45 /*
46 N = N0  $\wedge$  T = T0  $\wedge$   $0 \leq inf \leq sup \leq i \leq N - 1 \wedge$  *min_pos = MINPOS(T, 0, i)
47  $\wedge$  *max_pos = MAXPOS(T, 0, i)  $\wedge$  sum = SUM(T, i + 1, *min_pos, *max_pos)
48  $\wedge$  new_sum = NEWSUM(T, i + 2, *min_pos, *max_pos)  $\wedge$  T[i] < min
49 */
50 if(*max_pos <= *min_pos)
51 /*
52 N = N0  $\wedge$  T = T0  $\wedge$   $0 \leq inf \leq sup \leq i \leq N - 1 \wedge$  *min_pos = MINPOS(T, 0, i)
53  $\wedge$  *max_pos = MAXPOS(T, 0, i)  $\wedge$  sum = SUM(T, i + 1, *min_pos, *max_pos)

```

```

54       $\wedge$  new_sum = NEWSUM( $T, i + 2, *min\_pos, *max\_pos$ )  $\wedge T[i] < min$ 
55       $\wedge *max\_pos \leq *min\_pos$ 
56      */
57      sum += new_sum - T[*min_pos];
58      /*
59       $N = N_0 \wedge T = T_0 \wedge 0 \leq inf \leq sup \leq i \leq N - 1 \wedge *min\_pos = MINPOS(T, 0, i)$ 
60       $\wedge *max\_pos = MAXPOS(T, 0, i)$ 
61       $\wedge new\_sum = NEWSUM(T, i + 2, *min\_pos, *max\_pos) \wedge T[i] < min$ 
62       $\wedge *max\_pos \leq *min\_pos \wedge sum = SUM(T, i + 1, *min\_pos, *max\_pos) +$ 
63       $(new\_sum - T[*min\_pos])$ 
64      */
65
66      else
67      /*
68       $N = N_0 \wedge T = T_0 \wedge 0 \leq inf \leq sup \leq i \leq N - 1 \wedge *min\_pos = MINPOS(T, 0, i)$ 
69       $\wedge *max\_pos = MAXPOS(T, 0, i) \wedge sum = SUM(T, i + 1, *min\_pos, *max\_pos)$ 
70       $\wedge new\_sum = NEWSUM(T, i + 2, *min\_pos, *max\_pos) \wedge T[i] < min \wedge *max\_pos > *min\_pos$ 
71      */
72
73      sum = new_sum;
74      /*
75       $N = N_0 \wedge T = T_0 \wedge 0 \leq inf \leq sup \leq i \leq N - 1 \wedge *min\_pos = MINPOS(T, 0, i)$ 
76       $\wedge *max\_pos = MAXPOS(T, 0, i) \wedge new\_sum = NEWSUM(T, i + 2, *min\_pos, *max\_pos)$ 
77       $\wedge T[i] < min \wedge *max\_pos > *min\_pos \wedge sum = new\_sum$ 
78      */
79
80      new_sum = T[i];
81      /*
82       $N = N_0 \wedge T = T_0 \wedge 0 \leq inf \leq sup \leq i \leq N - 1 \wedge *min\_pos = MINPOS(T, 0, i)$ 
83       $\wedge *max\_pos = MAXPOS(T, 0, i) \wedge sum = SUM(T, i + 1, *min\_pos, *max\_pos)$ 
84       $\wedge new\_sum = T[i] \wedge T[i] < min$ 
85      */
86      min = T[i];
87      /*
88       $N = N_0 \wedge T = T_0 \wedge 0 \leq inf \leq sup \leq i \leq N - 1 \wedge *min\_pos = MINPOS(T, 0, i)$ 
89       $\wedge *max\_pos = MAXPOS(T, 0, i) \wedge sum = SUM(T, i + 1, *min\_pos, *max\_pos)$ 
90       $\wedge new\_sum = T[i] \wedge T[i] < min \wedge min = T[i]$ 
91      */
92      *min_pos = i;
93      /*
94       $N = N_0 \wedge T = T_0 \wedge 0 \leq inf \leq sup \leq i \leq N - 1$ 
95       $\wedge *max\_pos = MAXPOS(T, 0, i) \wedge sum = SUM(T, i + 1, i, *max\_pos)$ 
96       $\wedge new\_sum = T[i] \wedge T[i] < min \wedge min = T[i] \wedge *min\_pos = i$ 
97      */
98  }
99      if(T[i] > max){
100      /*
101       $N = N_0 \wedge T = T_0 \wedge 0 \leq inf \leq sup \leq i \leq N - 1 \wedge *min\_pos = MINPOS(T, 0, i)$ 
102       $\wedge *max\_pos = MAXPOS(T, 0, i) \wedge sum = SUM(T, i + 1, *min\_pos, *max\_pos)$ 
103       $\wedge new\_sum = NEWSUM(T, i + 2, *min\_pos, *max\_pos) \wedge T[i] > max$ 
104      */
105      if(*min_pos <= *max_pos)
106      /*
107       $N = N_0 \wedge T = T_0 \wedge 0 \leq inf \leq sup \leq i \leq N - 1 \wedge *min\_pos = MINPOS(T, 0, i)$ 
108       $\wedge *max\_pos = MAXPOS(T, 0, i) \wedge sum = SUM(T, i + 1, *min\_pos, *max\_pos)$ 
109       $\wedge new\_sum = NEWSUM(T, i + 2, *min\_pos, *max\_pos) \wedge T[i] > max$ 
110       $\wedge *min\_pos \leq *max\_pos$ 
111      */
112      sum += new_sum - T[*max_pos];
113      /*

```

```

114       $N = N_0 \wedge T = T_0 \wedge 0 \leq \text{inf} \leq \text{sup} \leq i \leq N - 1 \wedge *min\_pos = MINPOS(T, 0, i)$ 
115       $\wedge *max\_pos = MAXPOS(T, 0, i)$ 
116       $\wedge new\_sum = NEWSUM(T, i + 2, *min\_pos, *max\_pos) \wedge T[i] > max$ 
117       $\wedge *min\_pos \leq *max\_pos \wedge sum = SUM(T, i + 1, *min\_pos, *max\_pos) +$ 
118       $(new\_sum - T[*max\_pos])$ 
119       $*/$ 
120
121  else
122   $/*$ 
123       $N = N_0 \wedge T = T_0 \wedge 0 \leq \text{inf} \leq \text{sup} \leq i \leq N - 1 \wedge *min\_pos = MINPOS(T, 0, i)$ 
124       $\wedge *max\_pos = MAXPOS(T, 0, i) \wedge sum = SUM(T, i + 1, *min\_pos, *max\_pos)$ 
125       $\wedge new\_sum = NEWSUM(T, i + 2, *min\_pos, *max\_pos) \wedge T[i] > max \wedge *min\_pos > *max\_pos$ 
126       $*/$ 
127      sum = new_sum
128       $/*$ 
129       $N = N_0 \wedge T = T_0 \wedge 0 \leq \text{inf} \leq \text{sup} \leq i \leq N - 1 \wedge *min\_pos = MINPOS(T, 0, i)$ 
130       $\wedge *max\_pos = MAXPOS(T, 0, i) \wedge new\_sum = NEWSUM(T, i + 2, *min\_pos, *max\_pos)$ 
131       $\wedge T[i] > max \wedge *min\_pos > *max\_pos \wedge sum = new\_sum$ 
132       $*/$ 
133
134  new_sum = T[i];
135   $/*$ 
136       $N = N_0 \wedge T = T_0 \wedge 0 \leq \text{inf} \leq \text{sup} \leq i \leq N - 1 \wedge *min\_pos = MINPOS(T, 0, i)$ 
137       $\wedge *max\_pos = MAXPOS(T, 0, i) \wedge sum = SUM(T, i + 1, *min\_pos, *max\_pos)$ 
138       $\wedge new\_sum = T[i] \wedge T[i] > max$ 
139       $*/$ 
140  max = T[i];
141   $/*$ 
142       $N = N_0 \wedge T = T_0 \wedge 0 \leq \text{inf} \leq \text{sup} \leq i \leq N - 1 \wedge *min\_pos = MINPOS(T, 0, i)$ 
143       $\wedge *max\_pos = MAXPOS(T, 0, i) \wedge sum = SUM(T, i + 1, *min\_pos, *max\_pos)$ 
144       $\wedge new\_sum = T[i] \wedge T[i] > max \wedge max = T[i]$ 
145       $*/$ 
146  *max_pos = i;
147   $/*$ 
148       $N = N_0 \wedge T = T_0 \wedge 0 \leq \text{inf} \leq \text{sup} \leq i \leq N - 1 \wedge *min\_pos = MINPOS(T, 0, i)$ 
149       $\wedge sum = SUM(T, i + 1, *min\_pos, i)$ 
150       $\wedge new\_sum = T[i] \wedge T[i] > max \wedge max = T[i] \wedge *max\_pos = i$ 
151       $*/$ 
152  }
153  i++;
154   $/*$ 
155   $INV \equiv 0 \leq i \leq N \wedge T = T_0 \wedge N = N_0$ 
156   $\wedge *min\_pos = MINPOS(T, 0, i + 1)$ 
157   $\wedge *max\_pos = MAXPOS(T, 0, i + 1) \wedge sum = SUM(T, i + 2, *min\_pos, *max\_pos)$ 
158   $\wedge new\_sum = NEWSUM(T, i + 3, *min\_pos, *max\_pos)$ 
159   $*/$ 
160  }
161   $/*$ 
162   $INV \wedge \neg B \equiv i == N \wedge T = T_0 \wedge N = N_0$ 
163   $\wedge *min\_pos = MINPOS(T, 0, N)$ 
164   $\wedge *max\_pos = MAXPOS(T, 0, N) \wedge sum = SUM(T, N, *min\_pos, *max\_pos)$ 
165   $\wedge new\_sum = NEWSUM(T, N, *min\_pos, *max\_pos)$ 
166   $*/$ 
167  return sum;
168   $/*$ 
169   $POST \equiv T = T_0 \wedge N = N_0 \wedge sum = SUM(T, \text{inf}, \text{sup})$ 
170   $\wedge *max\_pos = MAXPOS(T, 0, N) \wedge *min = MINPOS(T, 0, N)$ 
171   $*/$ 
}

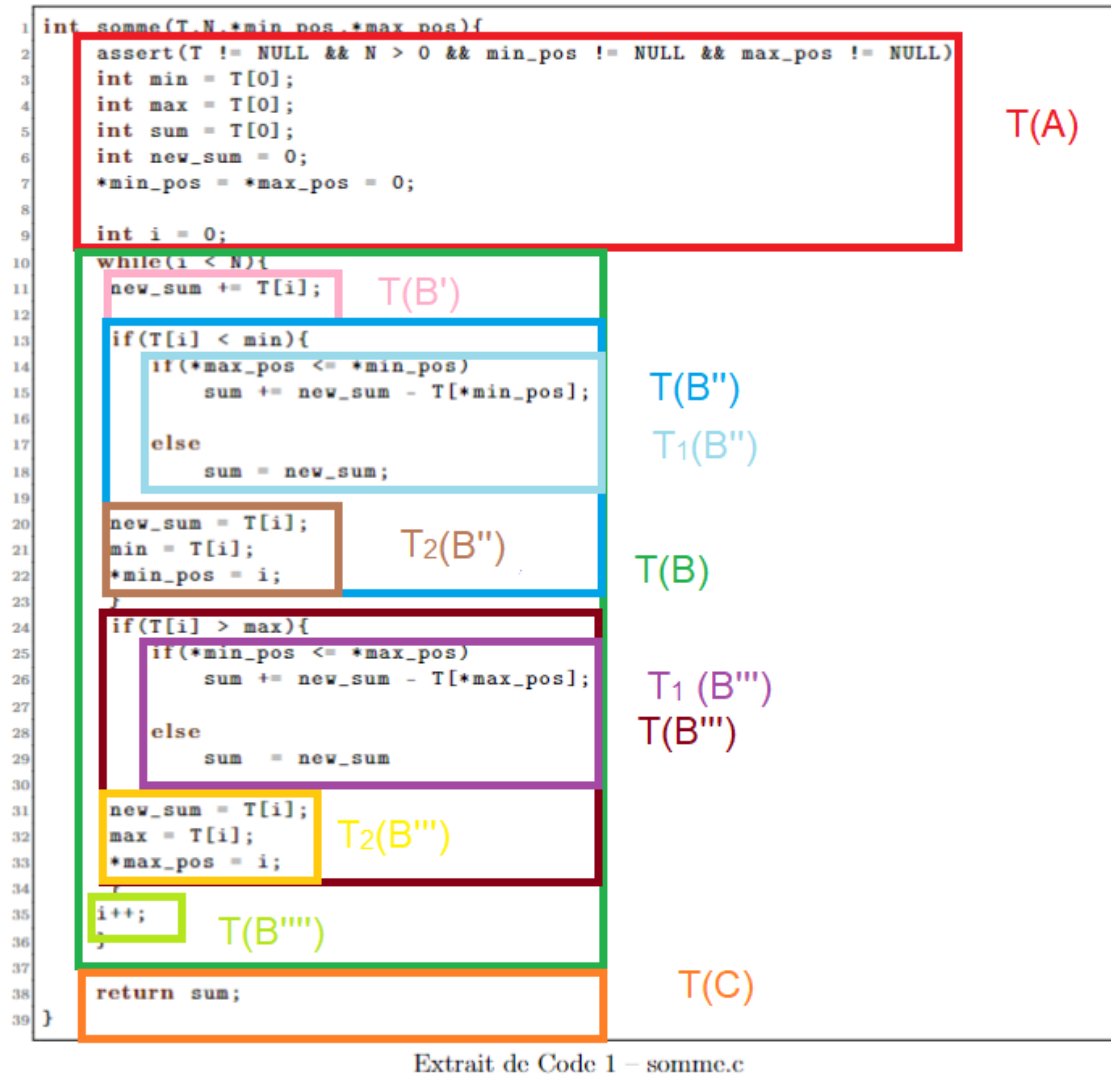
```


7 Code Complet

```
1 int somme(T,N,*min_pos,*max_pos){
2     assert(T != NULL && N > 0 && min_pos != NULL && max_pos != NULL)
3     int min = T[0];
4     int max = T[0];
5     int sum = T[0];
6     int new_sum = 0;
7     *min_pos = *max_pos = 0;
8
9     int i = 0;
10    while(i < N){
11        new_sum += T[i];
12
13        if(T[i] < min){
14            if(*max_pos <= *min_pos)
15                sum += new_sum - T[*min_pos];
16
17            else
18                sum = new_sum;
19
20            new_sum = T[i];
21            min = T[i];
22            *min_pos = i;
23        }
24        if(T[i] > max){
25            if(*min_pos <= *max_pos)
26                sum += new_sum - T[*max_pos];
27
28            else
29                sum = new_sum;
30
31            new_sum = T[i];
32            max = T[i];
33            *max_pos = i;
34        }
35        i++;
36    }
37
38    return sum;
39 }
```

Extrait de Code 1 – somme.c

8 Complexité



- Par application des règles 2 et 6 :

$$T(N) = T(A) + T(B) + T(C)$$

- Par application des règles 1 et 5 :

$$T(N) = 8 + \sum_{i=0}^{N-1} (T(B') + T(B'') + T(B''') + T(B''')) + 1$$

$$T(N) = 8 + \sum_{i=0}^{N-1} (1 + T(B'') + T(B''') + 1) + 1$$

- Quid de $T(B'')$?

$$T(B'') = T_1(B'') + T_2(B'')$$

- Quid de $T_1(B'')$? Par application de la règle 3 :

$$T_1(B'') = \max(T_{if}(), T_{else}())$$

$$T_{if}() = 1$$

et

$$T_{else}() = 1$$

donc

$$T_1(B'') = 1$$

- Quid de $T_2(B'')$? Par application de la règle 2 :

$$T_2(B'') = 3$$

- Ainsi $T(B'')$ devient :

$$T(B'') = 1 + 3$$

$$T(B'') = 4$$

- Quid de $T(B''')$?

$$T(B''') = T_1(B''') + T_2(B''')$$

- Quid de $T_1(B''')$? Par application de la règle 3 :

$$T_1(B''') = \max(T_{if}(), T_{else}())$$

$$T_{if}() = 1$$

et

$$T_{else}() = 1$$

donc

$$T_1(B''') = 1$$

- Quid de $T_2(B''')$? Par application de la règle 2 :

$$T_2(B''') = 3$$

- Ainsi $T(B''')$ devient :

$$T(B''') = 1 + 3$$

$$T(B''') = 4$$

- Ainsi $T(B)$ devient :

$$T(B) = \sum_{i=0}^{N-1} 10 = 10N$$

Donc :

$$T(N) = 8 + 10N + 1 = 9 + 10N$$

Par quoi borner $T(N)$?

$$O(n)$$

complexité linéaire.

Preuve ?

But : Trouver une constante $c \in R^+$ et un seuil n_0 à partir duquel $T(n_0) \leq c \times n_0$.

On remarque que pour $c = 11$ et $n_0 = 9$ on a $\forall n \geq n_0 : c \times n \geq 9 + 10N$

exemple :

- si $n = 9 : 99 \geq 99$
- si $n = 10 : 110 \geq 109$
- si $n = 11 : \dots$

9 Conclusion

Il est donc possible de résoudre ce problème en $O(N)$ (i.e en une seule boucle)