ÉCOLE POLYTECHNIQUE – ÉPREUVE FACULTATIVE D'ALGORITHMIQUE 2002

Solution avec Mathematica

PRÉLIMINAIRE

Q1

On utilise un parcours linéaire du tableau a pour déterminer le max et l'inf, la fonction est donc en temps linéaire ($\approx 2n$).

```
amplitude[a_] := Module[{max = -Infinity, min = Infinity},
Do[If[a[i] > max, max = a[i]];
   If[a[i] < min, min = a[i]],
{i, 0, n - 1}]; max - min]
```

Q2

Si le max précède le min, on aura du mal à faire un gain égal à amplitude ! Ex $a = \{100, 20\}$.
L'amplitude est la valeur absolue du gain ou de la perte maximale.

Q3

On utilise un parcours sur i, puis sur $j \leq i$, la fonction est donc en temps quadratique ($\approx n^2/2$).

```
gain[a_] := Module[{bid = 0},
 Do[If[a[j] - a[i] > bid, bid = a[j] - a[i]],
 {j, 1, n - 1}, {i, 0, j}]; bid]
```

Q4

La condition "j-i minimal" entraîne une nette complication. Une solution est d'inverser le sens de parcours des boucles imbriquées et d'introduire une variable et un test supplémentaires.

```
    gain[a_] := Module[{bid = 0, iMax, jMax,
attente = n},
    Do[
     If[a[j] - a[i] > bid, bid = a[j] - a[i];
                   attente = j-i;iMax = i; jMax = j];

        If[a[j] - a[i] == bid && (j-i<attente),
                   attente = j-i;iMax = i; jMax = j],

      {i, n - 1, 0, -1}, {j, i + 1, n - 1}];
     {bid, {iMax, jMax}}
     ]
```

Q5

Pour améliorer la complexité de l'algorithme, on va garder trace des indices des valeurs optimales.

```
gain1[a_] := Module[{gc = 0, indMax = 0, indMin = 0, nveauMin = 0},
  Do[If[a[i] > a[indMax], gc = a[indMax = i] - a[indMin]];
     If[a[i] - a[nveauMin] > gc, gc = a[indMax = i] - a[indMin = nveauMin]];
     If[a[i] < a[indMin], nveauMin = i]; ,
   {i, 1, n - 1}];
    gc]
```

Q6

simile:

```
gain1[a_] := Module[{gc = 0, indMax = 0, indMin = 0, nveauMin = 0},
  Do[If[a[i] > a[indMax], gc = a[indMax = i] - a[indMin]];
     If[a[i] - a[nveauMin] > gc, gc = a[indMax = i] - a[indMin = nveauMin]];
     If[a[i] < a[indMin], nveauMin = i]; ,
   {i, 1, n - 1}];
{gc, indMin,indMax}]
```

Q7

La solution la plus lisible utilise une coupure des fonctions précédentes, en les appliquant aux deux sous-tableaux obtenus en coupant a au point k et en optimisant la somme des deux par itération sur k.

Pour cela, on réécrit la procédure pour y faire figurer les dates extrêmes des transactions possibles.

```
gainCourtTerme[a_, dbut_, fin_] :=
Module[{gc = dbut, indMax = dbut, indMin = dbut, nveauMin = dbut},
     Do[If[a[i] > a[indMax], gc = a[indMax = i] - a[indMin]];
     If[a[i] - a[nveauMin] > gc, gc = a[indMax = i] - a[indMin = nveauMin]];
     If[a[i] < a[indMin], nveauMin = i];
         , {i, dbut + 1, fin - 1}];
gc]

gain2[a_] := Module[{essai, top = 0},
      Do[essai = gainCourtTerme[a, 0, k] + gainCourtTerme[a, k, n - 1];
          If[essai > top, top = essai], {k, 1, n - 1}];
 top]
```

Q8

Idem mais moins lisible:

```
gct[a_, dbut_, fin_] :=
    Module[{gc = dbut, indMax = dbut, indMin = dbut, nveauMin = dbut},
    Do[If[a[i] > a[indMax], gc = a[indMax = i] - a[indMin]];
       If[a[i] - a[nveauMin] > gc, gc = a[indMax = i] - a[indMin = nveauMin]];
       If[a[i] < a[indMin], nveauMin = i]; ,
    {i, dbut + 1, fin - 1}];
    {gc, indMin, indMax}]

    gain2[a_] := Module[{essai, coord, bid1, bid2, top = 0},
    Do[essai = First[bid1 = gct[a, 0, k]] + First[bid2 = gct[a, k, n - 1]];
    If[essai > top, top = essai;
                    coord = {bid1[[2]], bid1[[3]], bid2[[2]], bid2[[3]]}],
{k, 1, n - 1}];
    {top, coord}]
```

COMMENTAIRE

Voici une version plus idiosyncrasique où l'on adresse une liste par le numéro de ses éléments…ce qui permet de passer la liste en argument.

```
gct[a_List, debut_, fin_] :=
      Module[{gc = debut, indMax = debut,  indMin = d\'ebut, nveauMin = debut},
      Do[If[a[[i]] > a[[indMax]], gc = a[[indMax = i]] - a[[indMin]]];
      If[a[[i]] - a[[nveauMin]] > gc,
               gc = a[[indMax = i]] - a[[indMin = nveauMin]]];
      If[a[[i]] < a[[indMin]], nveauMin = i],
      {i, d\'ebut + 1, fin}];
      {gc, indMin, indMax}])

gain2[a_] := Module[{essai, coord, bid1, bid2, top = 0, n = Length[a]},
       Do[essai = First[bid1 = gct[a, 1, k]] + First[bid2 = gct[a, k, n]];
           If[essai > top, top = essai;
                   coord = {bid1[[2]], bid1[[3]], bid2[[2]], bid2[[3]]}],
       {k, 1, n - 1}];
      {top, coord}]
```