

Algorithme Min-Max pour le calcul du gain. Corrigé

1) def max(L) :

```
m = L[0]
for x in L :
    if x > m :
        m = x
return m
```

2) def Gain(G,P,E,s0) :

```
def traite(s) :    ### définit E[s] s'il ne l'est pas déjà
    if not (s in E) :
        for t in G[s] :
            if not (t in E) :
                traite(t)
    if P[s] :
        E[s] = max([E[t] for t in G[s]])
    else :
        E[s] = min([E[t] for t in G[s]])
return E[s0]
```

3) def Gain(G,P,E,s0) :

```
pile = [s0]
while pile :
    s = pile.head()
    if (s in E) :    ### permet un coût O(1) si E[s] a déjà été calculé
        s = pile.pop()
    else :
        flag = True    ### teste si E[t] a été calculé pour les successeurs de s
        for t in G[s] :
            if (t not in E) :
                pile.append(t) ; flag = False
        if flag :    ### ce bloc est effectuée au plus une fois pour chaque sommet
            s = pile.pop()
            if P[s] :
                E[s] = max([E[t] for t in G[s]])
            else :
                E[s] = min([E[t] for t in G[s]])
return E[s0]
```