Algorithm 1: Bottom-Up Budgeting

input : $Frame$: current frame
input : $Budget_{Frame}$: frame budget

```
1  for each CTU in Frame do
2  |  set(PS20, CTU);
3  end
4  Estimated_Comp ← updateEstimation();
5  Promote_PSet ← PS20;
6  while Estimated_Comp > Budget_Frame do
7  |  for each CTU with Promote_PSet in Frame do
8  |  |  promote(CTU);
9  |  end
10 |  Estimated_Comp ← updateEstimation();
11 |  Promote_PSet ++;
12 end
```

Algorithm 2: Priority-Based Budgeting

input : $Frame$: current frame
input : $Budget_{Frame}$: frame budget

```
1  for each CTU in Frame do
2  |  switch CTU_Depth do
3  |  |  case 4 set(PS100, CTU);
4  |  |  case 1 set(PS20, CTU);
5  |  |  case others set(PS60, CTU);
6  |  endsw
7  end
8  Estimated_Comp ← updateEstimation();
   // Refinement in case there is more/less budget
9  Demote_PSet ← PS100;
10 while Estimated_Comp < Budget_Frame do
11 |  for each CTU with Demote_PSet in Frame do
12 |  |  demote(CTU);
13 |  end
14 |  Estimated_Comp ← updateEstimation(PSx);
15 |  Demote_PSet −−;
16 end
17 Promote_PSet ← PS20;
18 while Estimated_Comp < Budget_Frame do
19 |  for each CTU with Demote_PSet in Frame do
20 |  |  promote(CTU);
21 |  end
22 |  Estimated_Comp ← updateEstimation();
23 |  Promote_PSet ++;
24 end
```