Задача №1 . Реализовать Complete greedy algorithm[пункт 2.4 .1 из Schreiber - Optimal Multi - Way NPP] с возможностью для пользователя выбрать вариант расчета :

In[•]:=

1) вернуть первое найденное решение;

In[•]:=
```
placeInSmaller[x_, {gr1_, gr2_}] := If[
  Total@gr1 > Total@gr2,
  {gr1, Join[gr2, x]},
  {Join[gr1, x], gr2}]
```

In[•]:=
```
placeInBigger[x_, {gr1_, gr2_}] := If[
  Total@gr1 < Total@gr2,
  {gr1, Join[gr2, x]},
  {Join[gr1, x], gr2}]
```

In[•]:=
```
Clear[cga1]
cga1[myset_, {gr1_, gr2_}] :=
 If [myset == {},
  {gr1, gr2},
  If[Length[myset] == 1,
   placeInSmaller[{First[myset]}, {gr1, gr2}],
   If[(Abs[Total@gr1 - Total@gr2] ≥ Total[myset]),
    placeInSmaller[myset, {gr1, gr2}],
    If[Total@gr1 == Total@gr2,
     cga1[Rest@myset, {Append[gr1, First[myset]], gr2}],
     cga1[Rest@myset, placeInSmaller[{First[myset]}, {gr1, gr2}]
     ]]]]]
```

In[•]:=
```
s = {10, 7, 6, 2}; (*пример с пары*)
cga1[s, {{}, {}}]
```

Out[•]= {{10, 2}, {7, 6}}

2) вернуть оптимальное решение;

In[•]:=
```
Clear[cga2]
cga2[myset_, {gr1_, gr2_}] :=
 If [myset == {},
  {gr1, gr2},
  If[Length[myset] == 1,
   placeInSmaller[{First[myset]}, {gr1, gr2}],
   If[(Abs[Total@gr1 - Total@gr2] ≥ Total[myset]),
    placeInSmaller[myset, {gr1, gr2}],
    If[Total@gr1 == Total@gr2,
     cga2[Rest@myset, {Append[gr1, First[myset]], gr2}],
     MinimalBy[
       {cga2[Rest@myset, placeInSmaller[{First[myset]}, {gr1, gr2}]],
        cga2[Rest@myset, placeInBigger[{First[myset]}, {gr1, gr2}]]},
       Abs[Total[#[[1]]] - Total[#[[2]]] &][[1]]
     ]]]]
```

In[•]:=
```
s = {10, 7, 6, 2}; (*пример с пары*)
cga2[s, {{}, {}}]
```

Out[•]=
```
{{10, 2}, {7, 6}}
```

3) вернуть наилучшее найденное решение за t секунд (t определяет пользователь) .

In[•]:=
```
Clear[cga3]
cga3[myset_, {gr1_, gr2_}] :=
 If [myset == {},
  Sow@{gr1, gr2},
  If[Length[myset] == 1,
   Sow@placeInSmaller[{First[myset]}, {gr1, gr2}],
   If[(Abs[Total@gr1 - Total@gr2] ≥ Total[myset]),
    Sow@placeInSmaller[myset, {gr1, gr2}],
    If[Total@gr1 == Total@gr2,
     cga3[Rest@myset, {Append[gr1, First[myset]], gr2}],
     MinimalBy[
       {cga3[Rest@myset, placeInSmaller[{First[myset]}, {gr1, gr2}]],
        cga3[Rest@myset, placeInBigger[{First[myset]}, {gr1, gr2}]]},
       Abs[Total[#[[1]]] - Total[#[[2]]] &][[1]]
     ]]]]
```

```
In[•]:= s = {10, 7, 6, 2};(*пример с пары*)
        time = 0.000000000000000000000001;
        result = Flatten[Reap[TimeConstrained[cga3[s, {{}, {}}], time]][2], 1];
        MinimalBy[result, Abs[Total[#[1]] - Total[#[2]]] &][1]
Out[•]= {{10, 2}, {7, 6}}
```

Объединение в одну функцию

```
In[•]:= completeGreedyAlgorithm[multiset_,
          variant_, time_ : 0.000000000000000000000001] := Module[
          {revmultiset = ReverseSort[multiset]},
          Which[variant == 1, cga1[revmultiset, {{}, {}}],
           variant == 2, cga2[revmultiset, {{}, {}}],
           variant == 3,
           MinimalBy[Flatten[Reap[TimeConstrained[cga3[revmultiset, {{}, {}}], time]][2], 1],
             Abs[Total[#[1]] - Total[#[2]]] &][1]]]


In[•]:= s = {10, 7, 6, 2};(*пример с пары*)
        completeGreedyAlgorithm[s, 1]
        completeGreedyAlgorithm[s, 2]
        completeGreedyAlgorithm[s, 3]
Out[•]= {{10, 2}, {7, 6}}

Out[•]= {{10, 2}, {7, 6}}

Out[•]= {{10, 2}, {7, 6}}

In[•]:= s = RandomInteger[{0, 20}, 6]
        completeGreedyAlgorithm[s, 1]
        completeGreedyAlgorithm[s, 2]
        completeGreedyAlgorithm[s, 3]
Out[•]= {17, 20, 16, 4, 19, 3}

Out[•]= {{20, 16, 4}, {19, 17, 3}}

Out[•]= {{20, 16, 4}, {19, 17, 3}}

Out[•]= {{20, 16, 4}, {19, 17, 3}}
```

In[ ]:=
```
s = RandomInteger[{0, 25}, 15]
completeGreedyAlgorithm[s, 1]
completeGreedyAlgorithm[s, 2]
completeGreedyAlgorithm[s, 3, 0.00000000000000000000000000000000000000000001]
```

Out[ ]= {2, 21, 5, 8, 4, 11, 9, 12, 1, 16, 13, 24, 13, 0, 21}

Out[ ]= {{24, 16, 13, 12, 9, 5, 1, 0}, {21, 21, 13, 11, 8, 4, 2}}

Out[ ]= {{24, 16, 13, 12, 9, 5, 1, 0}, {21, 21, 13, 11, 8, 4, 2}}

Out[ ]= {{24, 16, 13, 12, 9, 5, 1, 0}, {21, 21, 13, 11, 8, 4, 2}}

In[ ]:=