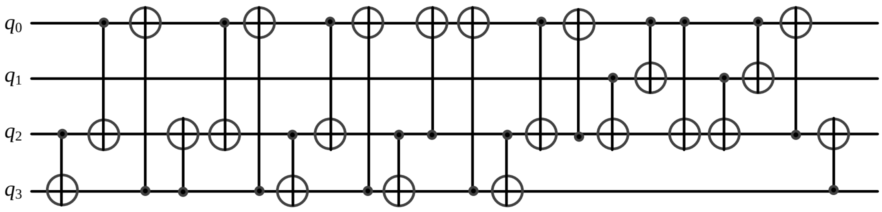
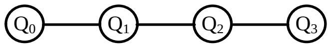
An example of Algorithm 4:

Consider the quantum circuit and the linear architecture in the figures below, and determine the optimal qubit allocation solutions as well as corresponding minimal routing cost according to Algorithm 4.





The logical qubit interaction frequency matrix **F** is:

0	2	8	4
2	0	2	0
8	2	0	6
4	0	6	0

The physical qubit routing distance matrix **R** is:

0	0	1	2
0	0	0	1
1	0	0	0
2	1	0	0

Initially, the problem size is 4, the lower bound is set to 0, the minimum cost is set to 12, the partial solution is empty, and the sorting array is set to $\{p_2, p_0, p_3, p_1\}$.

The cost matrix is:

0	*	*	*	*	0	*	*	*	*	0	*	*	*	*	0
*	0	2	4	0	*	0	2	2	0	*	0	4	2	0	*
*	0	8	16	0	*	0	8	8	0	*	0	16	8	0	*
*	0	4	8	0	*	0	4	4	0	*	0	8	4	0	*
*	0	2	4	0	*	0	2	2	0	*	0	4	2	0	*
0	*	*	*	*	0	*	*	*	*	0	*	*	*	*	0
*	0	2	4	0	*	0	2	2	0	*	0	4	2	0	*
*	0	0	0	0	*	0	0	0	0	*	0	0	0	0	*
*	0	8	16	0	*	0	8	8	0	*	0	16	8	0	*
*	0	2	4	0	*	0	2	2	0	*	0	4	2	0	*
0	*	*	*	*	0	*	*	*	*	0	*	*	*	*	0
*	0	6	12	0	*	0	6	6	0	*	0	12	6	0	*
*	0	4	8	0	*	0	4	4	0	*	0	8	4	0	*
*	0	0	0	0	*	0	0	0	0	*	0	0	0	0	*
*	0	6	12	0	*	0	6	6	0	*	0	12	6	0	*
0	*	*	*	*	0	*	*	*	*	0	*	*	*	*	0

The leader matrix is:

0	0	0	0
0	0	0	0
0	0	0	0
0	0	0	0

Invoke Algorithm2 as follows:
recu_qalloc(**C**, *size*=4, *min_cost*=12, *lb*=0, *pal_map*={}, *par_size*=0)

Then the permutation tree is traversed as follows:
1. The first visited node:
The root node is the visited first.
The lower bound and upper bound of the current cost matrix **C** are calculated according to Algorithm 1. *tem_l*=11, *tem_u*=12. After then, the cost matrix becomes:

0	*	*	*	*	0	*	*	*	*	0	*	*	*	*	0
*	3	0	0	0	*	1	0	0	0	*	0	0	0	1	*
*	0	2	8	2	*	0	4	4	0	*	2	8	3	0	*
*	0	0	2	0	*	0	0	1	0	*	0	2	0	0	*
*	0	0	1	0	*	0	0	2	0	*	0	2	0	0	*
0	*	*	*	*	0	*	*	*	*	0	*	*	*	*	0
*	0	0	2	2	*	0	3	1	0	*	2	1	0	0	*
*	0	0	0	0	*	0	0	0	0	*	0	0	0	0	*
*	0	1	6	1	*	0	5	5	0	*	1	5	1	0	*
*	5	0	0	0	*	0	0	0	1	*	0	0	0	4	*
0	*	*	*	*	0	*	*	*	*	0	*	*	*	*	0
*	0	0	2	0	*	0	3	2	0	*	0	2	0	0	*
*	0	0	3	0	*	0	2	2	0	*	0	1	0	0	*
*	2	0	0	0	*	1	0	0	0	*	1	0	0	4	*
*	0	3	6	0	*	0	2	3	0	*	0	5	1	0	*
0	*	*	*	*	0	*	*	*	*	0	*	*	*	*	0

The leader matrix becomes:

3	0	0	4
0	3	3	0
6	0	0	6
0	1	0	1

(1) The minimum cost is updated. *min_cost*= *lb*+*tem_u*=12. The feasible solution corresponding to this minimum cost is {2, 3, 1, 0}, and this solution is inserted into solution set *mapping*.

(2) The lower bound is updated. *lb*=*lb*+ *tem_l*=11.

(3) Allocate logical qubit p_2 (*sort_arr*[0])to physical qubit q_0, then *par_map*={$p_2 \rightarrow q_0$}, *par_size*++.

- Since *lb*+ L[2][0]=11+6=17 > *min_cost*, this assignment cannot produce better solution, so skip to the next iteration.

(4) Allocate logical qubit p_2 (*sort_arr*[0])to physical qubit q_1, then *par_map*={$p_2 \rightarrow q_1$}, *par_size*++.

- Since *lb*+ L[2][1]=11 < *min_cost*, update lower bound, *lb*=*lb*+ L[2][1]=11

- size--, size becomes 2.
- Reduce the cost matrix **C** and leader matrix **L**. The two matrices become:

0	*	*	*	0	*	*	*	0
*	0	0	0	*	0	0	1	*
*	0	2	1	*	0	2	0	*
*	0	1	2	*	0	2	0	*
0	*	*	*	0	*	*	*	0
*	0	0	0	*	0	0	0	*
*	0	3	2	*	0	1	0	*
*	0	0	0	*	1	0	4	*
0	*	*	*	0	*	*	*	0

4	0	12
0	3	0
0	0	5

Invoke Algorithm2 recursively. recu_qalloc(**C**, *size*=3, *min_cost*=12, *lb*=11, *par_map*={$p_2 \rightarrow q_1$}, *par_size*=1) (The steps are shown in "2. The Second visited node")

(5) Allocate logical qubit p_2 (*sort_arr*[0]) to physical qubit q_2, then *parital_map*={$p_2 \rightarrow q_2$}, *par_size*++.

- Since *lb*+ L[2][2]=11 < *min_cost*, update lower bound, *lb*=*lb*+ L[2][2]=11
- size--, size becomes 2.
- Reduce the cost matrix **C** and leader matrix **L**. The two matrices become:

0	*	*	*	0	*	*	*	0
*	3	0	0	*	0	0	0	*
*	0	2	0	*	0	2	0	*
*	0	1	0	*	0	2	0	*
0	*	*	*	0	*	*	*	0
*	0	0	0	*	0	0	0	*
*	0	3	0	*	2	1	0	*
*	2	0	0	*	0	0	0	*
0	*	*	*	0	*	*	*	0

10	0	5
0	4	0
5	1	0

- Invoke Algorithm2 recursively. recu_qalloc(**C**, *size*=3, *min_cost*=12, *lb*=11, *par_map*={$p_2 \rightarrow q_2$}, *pal_size*=1) (The steps are shown in "3. The Third visited node")

(6) Allocate logical qubit p_2 (*sort_arr*[0]) to physical qubit q_3, then *par_map*={$p_2 \rightarrow q_3$}, *par_size*++.

- Since *lb*+ L[2][3]=11+6=17 > *min_cost*, this assignment cannot produce better solution, so skip to the next iteration.

(7) In the end. The minimal routing cost is 12, and two optimal solutions are found, including {$p_0 \rightarrow q_2, p_1 \rightarrow q_3, p_2 \rightarrow q_1, p_3 \rightarrow q_0$} and {$p_0 \rightarrow q_1, p_1 \rightarrow q_0, p_2 \rightarrow q_2, p_3 \rightarrow q_3$}

2. The Second visited node:

Corresponding to the invocation recu_qalloc(**C**, *size*=3, *min_cost*=12, *lb*=11, *par_map*={$p_2 \rightarrow q_1$}, *par_size*=1)

(1) The lower bound and upper bound of the current cost matrix **C** are calculated according to Algorithm 1. *tem_l*=1, *tem_u*=1.

(2) Since *min_cost*== *lb*+*tem_l*=12. The complete solution corresponding to this minimum cost is {2, 3, 1, 0}, and this solution is inserted into solution set *mapping*.

(3) Since *tem_l*==*tem_u*, tthe sub-problem represented by this node is solved, so return from this node.

3. The Third visited node:

Corresponding to the invocation recu_qalloc(**C**, *size*=3, *min_cost*=12, *lb*=11, *par_map*={$p_2 \rightarrow q_2$}, *pal_size*=1)

(1) The lower bound and upper bound of the current cost matrix **C** are calculated according to Algorithm 1. *tem_l*=1, *tem_u*=1.

(2) Since *min_cost*== *lb*+*tem_l*=12. The complete solution corresponding to this minimum cost is {1, 0, 2, 3}, and this solution is inserted into solution set *mapping*.

(3) Since *tem_l*==*tem_u*, tthe sub-problem represented by this node is solved, so return from this node.