**Consult Materials**

1. **Christofides algorithm**
2. <https://en.wikipedia.org/wiki/Christofides_algorithm>, from this wiki I know the meaning and steps of this algorithm, which is the core theory of my code.
3. <https://en.wikipedia.org/wiki/Minimum_spanning_tree>, this page contains the detailed information of minimum spanning tree, including the properties and several algorithms. I write the method *minimum\_spanning\_tree* with lots of help of this page.
4. <https://en.wikipedia.org/wiki/Matching_(graph_theory)>,

***Computing Minimum-Weight Perfect Matchings*,** by WILLIAMCOOK

From the wiki of graph perfect matching, I know the basic theoretical knowledge of perfect matching, then I read a paper about computing minimum weight perfect matchings. These materials contribute a lot to my *min\_weight\_m* method.

1. <https://en.wikipedia.org/wiki/Eulerian_path>,

<https://www.geeksforgeeks.org/eulerian-path-and-circuit/>,

<https://www.geeksforgeeks.org/mathematics-euler-hamiltonian-paths/>

Above three web pages describe clearly about the Eulerian path and Hamiltonian path. I understand the meaning of Hamiltonian path and know how to find it by these materials. Then, I implement my method *find\_eul\_path*.

1. **Multiplicative Weight Updates**

I tried, but I don’t know how to do it.

1. **Data Structures**

**3.1 Fibonacci Heaps**

a. <https://en.wikipedia.org/wiki/Fibonacci_heap>

From this page I know the definition and structure of Fibonacci heap. My code was written step by step followed the instruction in this article.

b. <https://www.geeksforgeeks.org/fibonacci-heap-insertion-and-union/>

I also got lots of help from this tutorial, especially in the content of insert and union functions.

**3.2 Hollow Heaps**

I tried, but I don’t know how to do it.