This is the multi-way number partitioning problem, which is NPC.
1. binary search, and let $f[i][j]$ denote the minimum total time of the j-th worker can have among valid solutions that use j workers to finish job subset i. $O(2^n \cdot n^2 \log U)$, and we can assume $U \leq 2^n$.
2. let $f[i][j]$ be a pair of integers, the first integer denote the minimum number of workers needed to finish job subset i, and the second denote the minimum total time of the last worker. $O(2^n \cdot n \log U)$.
3. binary search, suppose we want to decide whether the answer $\leq mid$. construct a vector $A = \langle a_0, \ldots, a_{2^n-1} \rangle$, where $a_i = 1$ if the subset i has sum $\leq mid$, otherwise $a_i = 0$. Use FWT to compute A^k, and $ans \leq mid$ iff $A^k[2^n - 1] > 0$. $O(2^n \cdot n \log U)$.

References