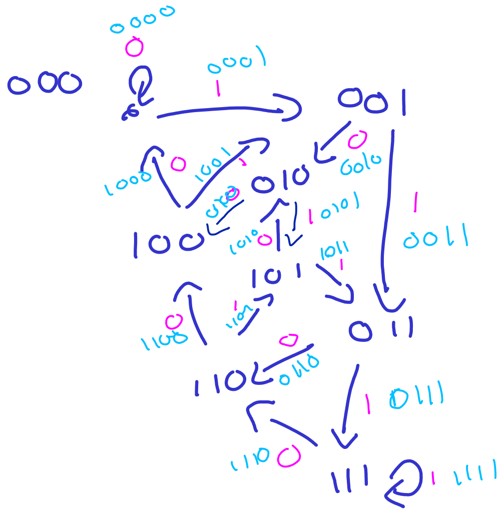
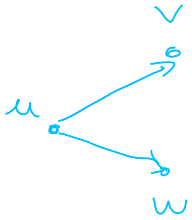
D_4



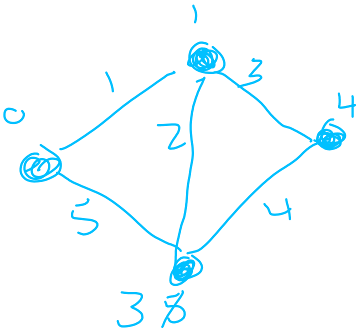


Algorithmic complexity

Complexity in time is a measure of how long we should perform steps in an algorithm for this algorithm to return an answer.

Dijkstra's algorithm

We visit one vertex, and for each vertex, we compute the distance to its neighbors.



# of steps for visiting vertices:

$$\sum_{v \in V} d(v) = 2|E|$$

# of steps for finding the minimum distance for non-visited vertices:

$$\sum_{i=0}^{n-1} (n-1)-i \sim \frac{n^2}{2}, \quad n = |V|$$

Notation of complexity:

$O(f(n))$: for an input of size n, the complexity is at most a multiple of the function f(n) (e.g. $f(n)=n^2$, 2^n, log(n), 3n+5, ...)

$\omega(f(n))$: for an input of size n, the complexity is at least a multiple of the function f(n)

$\Theta(f(n))$: for an input of size n, the complexity is roughly a multiple of the function f(n)