Master's Theorem

$T(n) = aT(n/b) + O(n^d)$

1. $d < \log_b a \Rightarrow T(n) = O(n^{\log_b a})$
2. $d = \log_b a \Rightarrow T(n) = O(n^d \log n)$
3. $d > \log_b a \Rightarrow T(n) = O(n^d)$

FFT

FFT(2^k length x) = FFT(x) = FFT(u+w, u−w)

where u = FFT(even coefficients)

w = ($w^0 v_0$, $w^1 v_1$, ...) where v = FFT(odd coeffs)

evaluate p(x) on $n = 2^k$ values $x_1, \dots x_n$ use n^{th} roots of unity.

FFT(ax + by) = a FFT(x) + b FFT(y)

SCCs alg:

- DFS on reverse graph, followed by DFS on original
- Discovers sink SCC's 1st then iteratively deletes sink components from graph.
- can topologically sort (linearize) SCCs of graph G

Edges by Post visit of DFS

edge(u,v) = back if pre(v) < pre(u) < post(u) < post(v)

edge(u,v) = forward if pre(u) < pre(v) < post(v) < post(u)

edge(u,v) = cross if pre(v) < post(v) < pre(u) < post(u)

Bellman Ford on DAG

$O(|V| + |E|)$ — to create linearized, then one pass of Bellman Ford

otherwise: Bellman Ford is $O(|V| \cdot |E|)$ if already in linearized order than $O(|V|)$

Kruskal's algorithm:

- finds MST: repeatedly add next lightest edge that doesn't make a cycle.

Property 1: removing cycle edge cannot disconnect a graph : $O(|E| \log |V|)$

Prim's algorithm:

- intermediate set of edges X always forms subtree and S is chosen to be set of this trees vertices
- very much like Djikstra's: doesn't care about cumulation though

Universal Hashing Family

- family of hash functions from A to B

$H = \{h : A \to B\}$ such that for any 2 elements $x \neq y \in A$

$Pr(h(x) = h(y)) = \frac{1}{|B|}$ where h is drawn uniformly at random from H.

Hash family: choose table size n to be some prime # a little larger than # items expected in table. Assume size of domain is $N = n^k$, then each data item can be considered as k tuple of integers modulo n and $H = \{h_a : a \in \{0, \dots n-1\}^k\}$ is universal family of hash functions

$\lim_{n \to \infty} \frac{f(n)}{g(n)} = \infty \Rightarrow f = \Omega(g)$ (f lowerbounded by g)

$\lim_{n \to \infty} \frac{f(n)}{g(n)} = 0 \Rightarrow f = O(g)$ (f upperbounded by g)

Runtime of DFS and BFS:

$O(|V| + |E|)$

Runtime of Djikstra's:

$O[(|V| + |E|) \log |V|]$ depending on choice of heap

d-ary heap runtime: $O\left[(|V|d + |E|)\frac{\log |V|}{\log d}\right]$

optimum choice of $d = \frac{|E|}{|V|}$

Adjacency list format: one list per vertex. iterating over vertex v's neighbors in O(degree(v)) time. takes up $O(|V| + |E|)$ space

Adjacency ~~matrix format~~: matrix with 1 at A_{ij} to represent edge $i \to j$

checking whether edge(u,v) in E in O(1) time.

$W = n{=}4$ FFT matrix

$$W^{-1} = \frac{1}{4}\begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & w^{-1} & w^{-2} & w^{-3} \\ 1 & w^{-2} & w^{-4} & w^{-6} \\ 1 & w^{-3} & w^{-6} & w^{-9} \end{bmatrix}$$