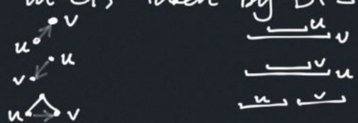
Generic DFS

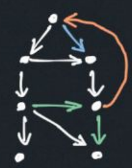
```
DFS G ((Σ, X), v) =
  if v ∈ X then (revisit (Σ, v), X)
  else let
     Σ' = visit (Σ, v)
     X' = X ∪ {v}
     (Σ'', X'') = iterate (DFS G) (Σ', X') N⁺(v)
  in
    (finish (Σ'', v), X'')
  end
```

DFSAll G Σ = iterate (DFS G) (Σ, ∅) V

Edge types

- Tree — in G, taken by DFS
- Back
- Forward
- Cross

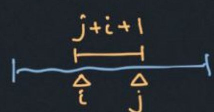




∃ Cycle ⇔ ∃ back edge in DFS

Quicksort

X_{ij} indicate i,j compared, $i<j$



$\mathbb{E}[X_{ij}] = \frac{1}{j-i+1} \underbrace{(2!)}_{\text{order to pick } i,j}$

$\mathbb{E}[\#\text{comparisons}] = \sum_{i<j} \mathbb{E}[X_{ij}] \leq 2 \sum_{i=0}^{n} H_n$

$\mathbb{P}[\text{ one path } > k \lg n] \leq \frac{1}{n^k}$

Topological Sort

Def $a < b \Leftarrow b$ reachable from $a \wedge b \neq a$

decreasingFinish given DAG:

```
Σ₀ = ⟨⟩
visit Σ v = Σ
finish Σ v = ⟨v⟩ ++ Σ
revisit Σ v = Σ
```

decreasingFinish G = #1 (DFSAll G Σ₀)

SCC, Kosaraju

Cost ~ 2× DFS

```
SCC G =
  let
     F = revFinishTime G
     Gᵀ = transpose G
                 visited verts  SCCs so far
    accumSCCs ((X, L), u) =
      let  overall visited
        (X', A) = reach Gᵀ X u
      in  newly visited     look for reachable, e.g. DFS
        if A = ∅ then (X, L) else
        (X', L ++ ⟨A⟩)
  in
    iterate accumSCCs (∅, ⟨⟩) F
  end
```

BFS

```
ParBFS G s =
  let explore X F i =
     if |F| = 0 then (X, i-1)
     else let
        X = X ∪ F
        F = N⁺_G(F) \ X
     in explore X F (i+1) end
  in explore ∅ {s} 0 end
```

Split

```
split (T, k) =
  case expose T of
    Leaf ⇒ (empty, false, empty)
   | Node (L, k', R) ⇒
     case cmp k k' of
       EQUAL ⇒ (L, true, R)
      | LESS ⇒ let (Lₗ, x, Lᵣ) = split (L, k)
               in (Lₗ, x, joinM(Lᵣ, k', R)) end
      ...
```

Sequences

$\langle i : 0 \leq i < n \mid i \% 2 = 0 \rangle$ — map/tab (↑ i), filter (↑ $i\%2=0$)

$A[1 \cdots 4]$ — inclusive

$A ++ B$ — append

reduce f b S usually $W(n) = 2W(\frac{n}{2}) + W_f(n)$

scan f b S usually $W(n) = W(\frac{n}{2}) + W_{contract}(n) + W_{expand}(n)$

scan append $\langle \langle m \rangle, \langle m \rangle, \ldots, \langle m \rangle \rangle$ (n)

	contract	recur	expand	solution
W	$2m \cdot \frac{n}{2} \in O(mn)$	$W(\frac{n}{2}, 2m)$	$O(n^2 m)$	$O(n^2 m)$
S	$O(1)$	$S(\frac{n}{2}, 2m)$	$O(1)$	$O(\lg n)$

Sorting $W = O(n \log n)$, $S = O(\log^2 n)$

scan op+ 0 ⟨2 1 1 4 3 2 1 3⟩

contract	⟨ 3	5	5	4	⟩
recur	(⟨ 0	3	8	13	⟩, 17)
expand	(⟨ 0 2	3 4	8 11	13 14	⟩, 17)

Log rules

$a^{k \log b} = b^{k \log a}$

$\log a^b = b \log a$

$a^{\log_a b} = b$

$\log_a b = \frac{\log_c b}{\log_c a}$

Summations

$\sum_{i=0}^{n} i^a \in \Theta(n^{a+1})$

$\sum_{i=0}^{n} \frac{1}{i} \in \Theta(\log n)$

$\sum_{i=0}^{n} b^i \in \Theta(b^n)$

$\sum_{i=0}^{\log n} \frac{n}{2^i} \in \Theta(n)$ $\quad \sum_{i=0}^{\log n} \frac{1}{2^i} \leq 2$

$\sum_{i=0}^{n} (\log^c i)(i^a)(b^i)$

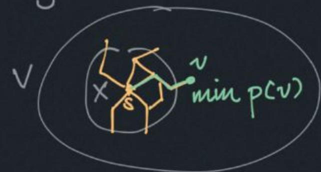
$\in \Theta\left((n \log^c n)(n^a)(b^n)\right)$ $\quad b \geq 1,\ a, c \geq 0$

Dijkstra

Prop $p(v) := \min_{u \in X} (\delta(s,u) + w(u+v))$

Then $v \in V \setminus X$ with $\min p(v)$ has $\delta(s,v) = p(v)$

viz. best way to expand seen set is greedily taking min way out



i.e. $\min_{v \in V \setminus X} p(v) = \min_{v \in V \setminus X} \delta(s,v)$

```
DijkstraPQ G s =
  let loop X Q =
    case delMin Q of
      (NONE, _) => X
    | (SOME (d,v), Q') =>
        if (v,_) ∈ X then loop X Q'
        else let
          X' = X ∪ {(v,d)}
          relax (Q,(u,w)) = insert (Q, (d+w, u))
          Q'' = iterate relax Q' N+G(v)
        in
          loop X' Q''
        end
  in
    loop {} (insert emptyQ (0,s))
  end
```

Tree pq impl $O(m \lg n) = O(m \lg m)$

Dijkstra alt with Fibb heap < $O(1)$ insert, update; $O(\lg n)$ delMin

$W = O(m + n \lg n)$

Exs.

SP by both weight and # edges

→ Dijkstra but tie break pq by both weight & edge dist

Have $\mathbb{E}X_0 = n$, $\mathbb{E}X_{i+1} = (\frac{1}{2})^i n$

Fix k, WTS done in $O(k \lg n)$ whp.

$r := (k+1) \log_{(\frac{1}{2})^{-1}} n$

$$P[X_r \geq 1] \leq \mathbb{E}X_r = \left(\frac{1}{2}\right)^{(k+1)\lg n} n = n^{(k+1)(-1)} n = \frac{1}{n^k}$$

Treaps

$P[A_j^i] = \frac{1}{|i-j|+1}$ ← need choose i first (i ancestor of j)

$E[\text{depth}(i)] = \sum_{i=0}^{n-1} E[A_j^i] \leq 2H_n$

$E[\text{size}(i)] \in O(\lg n)$

Def balanced : height $\in O(\lg n)$

Graph Search Summary

	W	S	
DFS	$m+n$ $\times \lg n$ if visit/revisit/finish are $O(\lg n)$	$m+n$	revisit max m, done in n
BFS	$(m+n)\lg n$ (set op)	$d \lg^2 n$ (depth; set ops $\lg n$ for tree)	
Dijkstra	$m \lg n$ or $(m+n)\lg n$	$m \lg n$	
BF	mn update all node dist for every edge	$n \lg n$ (par $O(\frac{n}{\lg n})$)	

Prob

Random var $X : \Omega \to \mathbb{R}$

Expected var $\mathbb{E}[X]$ = weighted sum

Indep X, Y indep $\Leftrightarrow P[X=a, Y=b] = P[X=a]P[Y=b]$

Linearity $\mathbb{E}[X+Y] = \mathbb{E}[X] + \mathbb{E}[Y]$

Product expectation $\mathbb{E}[X \cdot Y] = \mathbb{E}[X] \cdot \mathbb{E}[Y]$

Union bound $P(A \cup B) \leq P(A) + P(B)$

Conditional $P(A|B) = \frac{P(A \cap B)}{P(B)}$

Markov inequality $X \geq 0 \Rightarrow P[X \geq a] \leq \frac{\mathbb{E}[X]}{a}$

Prob bound

High Prob Bound $W(n) \in O(f(n))$ whp. if $\exists$ constants c, n_0 s.t. $\forall n > n_0, \forall k, P[W(n) \leq ckf(n)] \geq 1 - (\frac{1}{n})^k$

Event A has high prob scaled with n if $P[\neg A] \leq \frac{1}{n^k}$

Proving
1. $N_r :=$ input size at round r
2. Decay ratio $\frac{\mathbb{E}[N_r]}{\mathbb{E}[N_{r-1}]} \leq p$
3. $\mathbb{E}[N_r] \leq p^r N_0 = p^r n$
4. Set $r = (k+1)\log_{\frac{1}{p}} n$, show $\mathbb{E}[N_r] \leq \frac{1}{n^k}$

Partition cost $\mathbb{E}[T(n)] = \mathbb{E}[T(n) \mid \text{good}]P[\text{good}] + \cdots$

Bellemam-Ford

k-hop shortest, then relax every next hop. n-1 hops enough

```
BF G s =
  let loop (D:(V,R) table) k =
    let D' = {v ↦ min (D[v], min_{u∈N-G(v)} D[v]+w(v,u)) : v∈V}
    in
      if k=|V| then NONE
      else if D'=D then SOME D
      else loop D' (k+1)
  in
    loop {s ↦ 0} ∪ {v ↦ ∞ : v ∈ V\{s}} 0
  end
```

min cost to hop into vert from end of prev hops

Misc

$H_n = \sum_{i=1}^{n} \frac{1}{i} \leq \ln n + O(1)$

$\|F\| = \sum_{x \in F} (d^+(x) + 1)$

BST perfect balanced depth $= \lceil \lg(n+1) \rceil$