WRAP UP FFT

DFS FOR TOPOLOGICAL SORT

Feb 5, 2019

Polynomial multiplication

$$(1 + 2x + x^2) \cdot (2 + x + 3x^2)$$

$$= 2 + (2+1) \cdot x + (3+2+2)x^2 + (\;)x^3 + 3x^4$$

given $A(x) = a_0 + a_1 x + \cdots + a_n x^n$

$B(x) = b_0 + b_1 x + \cdots + b_n x^n$

want to find coefficients of

$$C(x) = A(x) \cdot B(x)$$

$$C(x) = c_0 + c_1 x + \cdots + c_{2n} x^{2n}$$

$c_0 = a_0 b_0$

$c_1 = a_0 b_1 + a_1 b_0$

$\Theta(n^2)$ $\quad$ $c_2 = a_2 b_0 + a_1 b_1 + a_0 b_2$

$\vdots$

$c_n = a_n b_0 + a_{n-1} b_1 + \cdots + a_0 b_n$

$c_{n+1} = a_n b_1 + \cdots$

given $A(x) = a_0 + a_1 x + \dots + a_n x^n$

$B(x) = b_0 + b_1 x + \dots + b_n x^n$

want to find coefficients of

$C(x) = A(x) \cdot B(x)$

$C(x) = c_0 + c_1 x + \dots + c_{2n} x^{2n}$

Inverse FFT $\quad$ C is a polynomial of degree $\leq N-1$ where N is a power of 2

given $C(1), C(w), \dots, C(w^{N-1})$

where $1, w, \dots, w^{N-1}$ are N-th roots of unity

in $O(N \log N)$ time find coefficients of C

---

given $A(1), A(w), \dots, A(w^{N-1})$

$B(1), B(w), \dots, B(w^{N-1})$

then in $O(N)$ time can compute

$C(1) = A(1) \cdot B(1), C(w) = A(w) B(w), \dots, C(w^{N-1}) =$
$= A(w^{N-1}) B(w^{N-1})$

---

with FFT given A, B

compute $A(1) \dots A(w^{N-1})$

$B(1) \dots B(w^{N-1})$ in $O(N \log N)$ time

Input $A(x) = a_0 + a_1 x + \cdots + a_n x^n$

$B(x) = b_0 + b_1 x + \cdots + b_n x^n$

Let N be a power of $2 \geq 2n+1$ and $\leq 4n$

$A(1), A(\omega), \ldots, A(\omega^{N-1}) = FFT(A, N)$

$B(1), B(\omega), \ldots, B(\omega^{N-1}) = FFT(B, N)$

$C(1) = A(1) \cdot B(1)$

$\vdots$

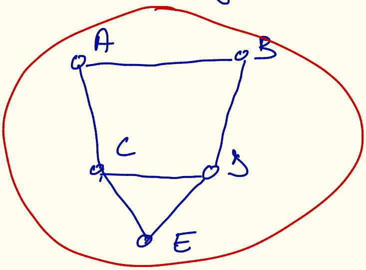
$C(\omega^{N-1}) = A(\omega^{N-1}) B(\omega^{N-1})$

$c_0 \, c_1 \ldots c_{N-1} = IFFT(C(1), \ldots, C(\omega^{N-1}))$

$$O(N \log N) = O(n \log n)$$

Graph

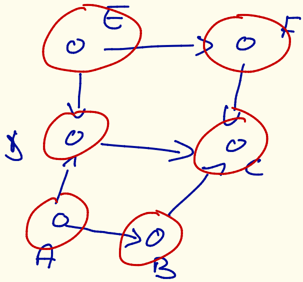
Def: undirected graph



Vertices = { A, B, C, D, E }

Edges = { (A,B), (A,C), ( ) - - }

directed graph



Vertices = { — }

Edges = { - — }

Path

A — B — D — E

E → D → C

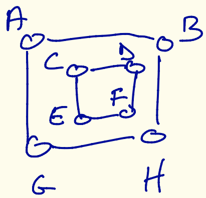
Reachability

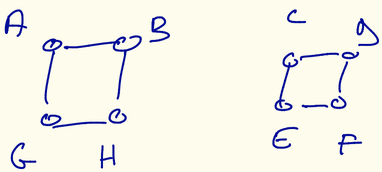
u —— v

u ——→ v

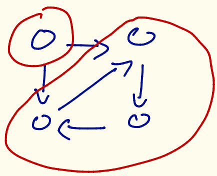
v is reachable from u

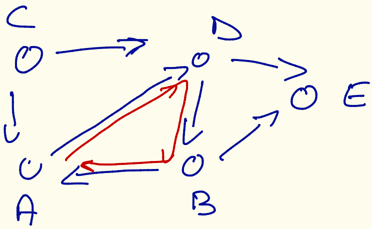
Connected component





Strongly connected component





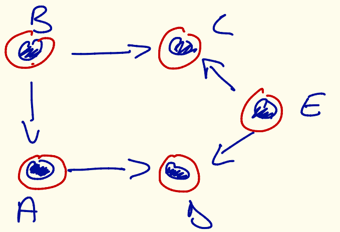
C D A B E

If a directed graph has a cycle
Then it does not a topological sort

visited = boolean array indexed by vertices initialized to F
L = empty list

```
def explore(v):
    visited[v] = T
    for each neighbor w of v
        if not visited[w]:
            explore(w)
    L = [v] + L
```

```
def DFS
    for each vertex v
        if not visited[v]: explore(v)
```



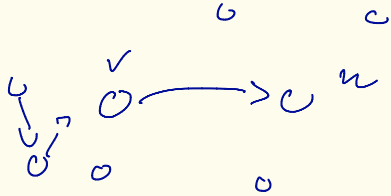
EBCAD

(1) Runs in time $O(|V| + |E|)$

Trace execution of algorithm
Consider nodes v in order in which explore(v) terminates
Reverse of that order is a topological sort if no cycles

Suppose G has no cycles

Algorithm outputs a valid topological sort



explore(v) is called when visited[w] = F

explore(w) is called inside explore(v)

explore(w) terminates before explore(v)

explore(w) is called when visited[v] = F

$\boxed{\text{can explore(v) be called inside execution of explore(w)? NO}}$

explore(v) is called (and so it terminates) after explore(w) terminates