Lecture 31: Friday November 23, 2001

today

- single source shortest path: Dijkstra
- divide and conquer: matrix multiplication

announcements

- Assignment 8 due *Wednesday* November 28

single-source shortest path problem

- distance(x,y): min'm sum of edge weights in x-y path $\qquad \infty$ if none
- given starting vertex s, find dist(s,v) for all vertices v

directed, weighted version (non-negative weights)

- solved by Dijkstra's algorithm
  - greedy, similar to Prim/Dijkstra/Boruvka MST algorithm
  - also works for undirected graphs
  - also works for unweighted (make all weights 1)

Dijkstra SSSP

```
Relax(u,v,w)        { shorter to v via (u,v)? { [CLRS p 586]
if d[v] >  d[u] + w(u,v) then
    d[v] <- d[u] + w(u,v)
    parent[v] <- u

InitSS(G,s)                          { [CLRS p 585]
for each vertex v in V[G] do
  d[v] <- infinity        { shortest (s,v) distance so far
  parent[v] <- NIL        { allows path construction
d[s] <- 0                 { s is source

Dijkstra(G,w,s)  { digraph, weights non-neg. [CLRS p595]
InitSS(G,s)
S <- empty     { vertices with d[v] final
Q <- V[G]      { min priority queue
while not isEmpty(Q) do
  u <- ExtractMin(Q)
  add u to S
  for each vertex v in Adj[u] do
    Relax(u,v,w)
```

correctness of Dijkstra SSSP

- claim: at termination, d[v] is dist(s,v) for each v in V
- loop invariant: at start of Line 4, d[v] is dist(s,v) for each v in S

proof of loop invariant

- init'n: S empty, so invariant holds vaccuously
- next step $S = \{s\}$, and d[s]=0=dist(s,s), so invariant holds
- maintenance ?
- term'n: S=V, so invariant implies claim

proof of loop invariant: maintenance

- only change to S in loop: add u
- so must show: at loop body end, d[u] is dist(s,u)

showing d[u] is dist(s,u)

- consider any shortest s-u path $P = (s = v_0, v_1, \ldots, v_k = u)$
- $y = v_j$: first vertex in P not in S — possibly $y = u$
- $x = v_{j-1}$ — possibly $x = s$
- d[y] = dist(s,y) when u added to S — why?
  - any subpath of a shortest path is a shortest path
  - so $(s, \ldots, x, y)$ is a shortest path
  - x is in S
  - so (x, y) was relaxed when $d[x] = dist(s, x)$
  - $d[v]$ never increases, so $d[x] = dist(s, x)$ still
  - so $d[y] \leq length(s, \ldots, y) = dist(s, y)$
- y precedes (or equals) u in shortest s, u path
- so dist(s,y) $\leq$ dist(s,u)
- y and u both in $V - S$ when u chosen, so $d[u] \leq d[y]$
- …but also $d[y] = dist(s, y) \leq dist(s, u) \leq d[u]$
- so $d[y] = dist(s, y) = dist(s, u) = d[u]$ ☺

Dijkstra SSSP run time

- same as PDB MST
- WC $\Theta(n^2)$ with priority queue: list imp'n
- WC $\Theta(m \lg n)$ with priority queue: heap imp'n

new problem: matrix multiplication

- definition of $C = A_{n\times m} * B_{m\times n}$

$$A = \begin{bmatrix} a_{11} & a_{12} & \dots & \dots & a_{1m} \\ a_{21} & a_{22} & \dots & \dots & a_{2m} \\ \dots & \dots & \dots & \dots & \dots \\ a_{n1} & a_{n2} & \dots & \dots & a_{nm} \end{bmatrix} \qquad B = \begin{bmatrix} b_{11} & b_{12} & \dots & b_{1n} \\ b_{21} & b_{22} & \dots & b_{2n} \\ \dots & \dots & \dots & \dots \\ \dots & \dots & \dots & \dots \\ b_{m1} & b_{m2} & \dots & b_{mn} \end{bmatrix}$$

$$C = \begin{bmatrix} c_{11} & c_{12} & \dots & c_{1n} \\ c_{21} & c_{22} & \dots & c_{2n} \\ \dots & \dots & \dots & \dots \\ c_{n1} & c_{n2} & \dots & c_{nn} \end{bmatrix} \qquad c_{jk} = \begin{bmatrix} a_{j1} & a_{j2} & \dots & a_{jm} \end{bmatrix} \begin{bmatrix} b_{1k} \\ b_{2k} \\ \dots \\ \dots \\ b_{mk} \end{bmatrix}$$

divide and conquer matrix multiplication

- partition A, B into 4 submatrices
  - assume $n = m$
  - $A = \begin{bmatrix} A_{11} & \mid & A_{12} \\ -- & & -- \\ A_{21} & \mid & A_{22} \end{bmatrix}$ and $B = \begin{bmatrix} B_{11} & \mid & B_{12} \\ -- & & -- \\ B_{21} & \mid & B_{22} \end{bmatrix}$
  - $C' \leftarrow \begin{bmatrix} A_{11} * B_{11} + A_{12} * B_{21} & \mid & A_{11} * B_{12} + A_{12} * B_{22} \\ ----------- & & ----------- \\ A_{21} * B_{11} + A_{22} * B_{21} & \mid & A_{21} * B_{12} + A_{22} * B_{22} \end{bmatrix}$
  - $C' = C$ (check: $c'_{jk} = c_{jk}$ for each j, k) ☺

analysis

- $A(n), M(n)$: total number of add'ns, mult'ns

$$M(n) = \begin{cases} 8M(n/2) & \text{if } n \geq 2 \\ 1 & \text{if } n = 1 \end{cases}$$

$$A(n) = \begin{cases} 8A(n/2) + n^2 & \text{if } n \geq 2 \\ 0 & \text{if } n = 1 \end{cases}$$

- $M(n)$: $a = 8$, $b = 2$, $c = 0$
  $a > b^c$ so $M(n) \in \Theta(n^{\lg_2 8}) = \Theta(n^3)$
- $A(n)$: $a = 8$, $b = 2$, $c = 2$
  $a > b^c$ so $A(n) \in \Theta(n^{\lg_2 8}) = \Theta(n^3)$
- conclusion: same as ordinary matrix mult'n ☹

Strassen's divide/conquer matrix mult'n [CLRS 28.2]

$$
\begin{aligned}
X_1 &\leftarrow (A_{11} + A_{22}) * (B_{11} + B_{22}) \\
X_2 &\leftarrow (A_{21} + A_{22}) * B_{11} \\
X_3 &\leftarrow A_{11} * (B_{12} - B_{22}) \\
X_4 &\leftarrow A_{22} * (B_{21} - B_{11}) \\
X_5 &\leftarrow (A_{11} + A_{12}) * B_{22} \\
X_6 &\leftarrow (A_{21} - A_{11}) * (B_{11} + B_{12}) \\
X_7 &\leftarrow (A_{12} - A_{22}) * (B_{21} + B_{22})
\end{aligned}
$$

$$
C^* \leftarrow \left[\begin{array}{c|c} X_1 + X_4 - X_5 + X_7 & X_3 + X_5 \\ \hdashline X_2 + X_4 & X_1 + X_3 - X_2 + X_6 \end{array}\right]
$$

- correctness: $C^* = C$ (exercise)

- analysis:

$$
\begin{aligned}
M(n) &= \begin{cases} 7M(n/2) & \text{if } n \geq 2 \\ 1 & \text{if } n = 1 \end{cases} \\
A(n) &= \begin{cases} 7A(n/2) + \frac{18}{4}n^2 & \text{if } n \geq 2 \\ 0 & \text{if } n = 1 \end{cases}
\end{aligned}
$$

  - exercise: give r.r. for $T(n) = A(n) + M(n)$
  - $M(n)$: $a = 7$, $b = 2$, $c = 0$
    $a > b^c$ so $M(n) \in \Theta(n^{\lg_2 7}) = \Theta(n^{2.81\ldots})$
  - $A(n)$: $a = 7$, $b = 2$, $c = 2$
    $a > b^c$ so $A(n) \in \Theta(n^{\lg_2 7}) = \Theta(n^{2.81\ldots})$
  - conclusion: better than ordinary ☺

divide/conquer matrix multiplication: conclusions

- ordinary $\Theta(n^3)$
- Strassen $\Theta(n^{2.81\cdots})$
- in practice
  - for small n ordinary (not d/c) beats Strassen
  - for larger n Strassen beats ordinary
- current best algorithm approx $\Theta(n^{2.376})$
- best lower bound (number of array elements) $\Omega(n^2)$
- how hard is matrix multiplication?
  - not sure (open problem) between these two bounds