Lecture 20: Wednesday October 24, 2001

today

- dynamic programming [CLRS Ch. 15.1-3]
  - chained matrix multiplication: analysis
- graphs (intro) [CLRS Appendix B.4]
- union find [CLRS Ch. 21]

announcements

- assignment 5: due Monday October 29
- acknowledge references
- feedback
  - slides (advance time)
  - groups

```
Matrix-Chain-Order(D)  [based on CLRS p 336]
1 n <- length(D)-1        { D[0..n]: array dimensions
2 for x <- 1 to n do
3    M[x,x] <- 0
4 for L <- 2 to n do      { L: current chain length
5    for x <- 1 to n-L+1 do
6       y <- x+L-1; M[x,y] <- infinity
7       for t <- x to y-1 do
8          q <- M[x,t] + M[t+1,y] + D[x-1]D[t]D[y]
9          if q < M[x,y] then
10            M[x,y] <- q; S[x,y] <- t
```

- space in $\Theta(n^2)$ why?
- run time prop'l to total number of line executions why?
- line 1? 1 lines 2,3? n lines 4,...,10?
- run time prop'l to $T(n)$, number of line 8,9 executions? why?
- $T(n) = \sum_{L=2}^{n} \sum_{x=1}^{n-L+1} \sum_{t=x}^{x+L-2} 1 \in \Theta(n^3)$?
- run time in $\Theta(n^3)$

$$
\begin{aligned}
T(n) &= \sum_{L=2}^{n} \sum_{x=1}^{n-L+1} \sum_{t=x}^{x+L-2} 1 \\
&= \sum_{L=2}^{n} \sum_{x=1}^{n-L+1} L-1 \\
&= \sum_{L=2}^{n} (n-L+1)(L-1) \\
&= (n-1)1 + (n-2)2 + (n-3)3 + \ldots + 2(n-2) + 1(n-1) \\
&\in \Theta(n^3)
\end{aligned}
$$

since

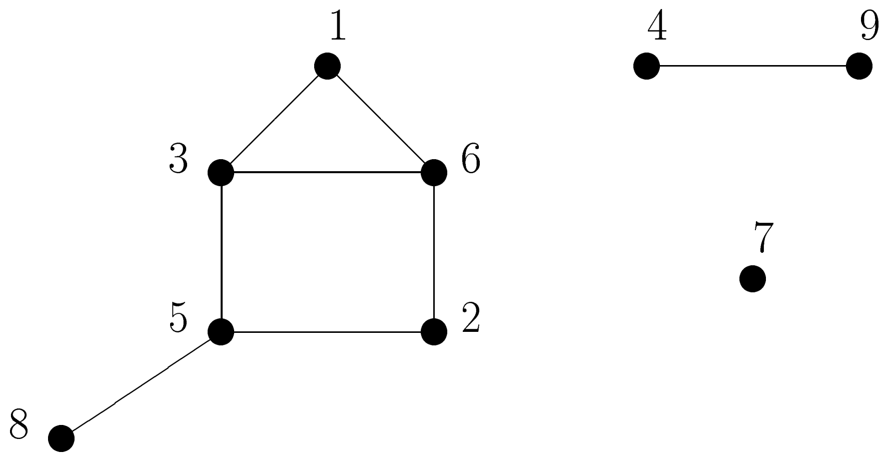
$$
\begin{aligned}
T(n) &\le \sum_{L=2}^{n} n^2 \ \text{ and} \\
T(n) &\ge \sum_{L=\lfloor n/3 \rfloor}^{\lfloor 2n/3 \rfloor} (n-L+1)(L-1) \\
&> \sum_{L=\lfloor n/3 \rfloor}^{\lfloor 2n/3 \lfloor} (\frac{n}{3}-2)(\frac{n}{3}-2) \\
&\ge \frac{n}{3}(\frac{n}{3}-2)(\frac{n}{3}-2) \\
&\in \Omega(n^3)
\end{aligned}
$$

other problems suited to dynamic programming

- constructing optimal binary search trees
- string matching: longest common subsequence
- all pairs shortest paths in (di)graph

next: graphs   [CLRS Appendix B.4]

- eventually . . .
  - definitions
  - basic properties
  - algorithms
- now . . .
  - brief introduction



1:	3 6
2:	5 6
3:	1 5 6
4:	9
5:	2 3 8
6:	1 2 3
7:	
8:	5
9:	4

	1	2	3	4	5	6	7	8	9
1			*			*			
2					*	*			
3	*				*	*			
4									*
5		*	*					*	
6	*	*	*						
7									
8					*				
9				*					

- (simple, undirected) graph $G = (V, E)$:
  vertex set V and edge set E consisting of unordered vertex pairs
- adjacent, incident, degree
- computer representation: adjacency lists; adjacency matrix
- path
- connected
- (connected) component

some graph definitions

- *adjacent* vertices induce an edge; *adjacent* edges share a vertex
- *incident* edge and vertex: if the vertex is in the edge
- vertex *neighbourhood*: all vertices adjacent to the vertex
- vertex *degree*: size of its neighbourhood
- in a graph, a vertex sequence $(v_0, v_1, \ldots, v_t)$ is a *path*

  if each consecutive pair of vertices is adjacent
- *connected* graph: for each pair of vertices, a path between them
- *subgraph* of $G = (V, E)$: a graph $G' = (V', E')$ with $V' \subseteq V$ and $E' \subseteq E$
- graph *component*: maximal connected subgraph
- binary *equivalence relation*: reflexive, symmetric, transitive
- *equivalence class*: maximal sets of equivalent pairs

  e.g. for any graph, the relation $R(x, y)$: *there is a path* $(x, \ldots, y)$ is an equivalence relation whose equivalence classes are the graph's components