Homework #6: dynamic programming, greedy algorithms, minimum spanning trees
Due Date: Tuesday, 28 May 2002

Reading: Chapters 16, 17, 24 in CLR, 15, 16, 23 in CLRS.

Recall that *exercises* are for you to work out on your own; *problems* are to be handed in.

Exercise 6-1. Do Exercise 16.3–4 on page 319 of CLR, 15.4–4 on page 356 of CLRS.

Exercise 6-2. Do Problem 16–2 on page 325 of CLR, 15–2 on page 364 of CLRS..

Exercise 6-3. Do Exercise 17.2–4 on page 337 of CLR, 16.2–4 on page 384 of CLRS.

Exercise 6-4. Do Exercise 24.2–4 on page 510 of CLR, 23.2–4 on page 574 of CLRS.

Exercise 6-5. Do Exercise 24.2–5 on page 510 of CLR, 23.2–5 on page 574 of CLRS.

Problem 6-1. Cutting a log [50 points]

You bring an ℓ–foot log of wood to your favorite sawmill. You want it cut in k specific places: $\ell_1, \ell_2, \ldots, \ell_k$ feet from the left end. The sawmill charges x dollars to cut an x–foot log any place you want.

(a) Give an efficient dynamic programming algorithm to determine the order in which they should cut your log in order to minimize your cost. The running time of your algorithm should be polynomial in k. Analyze the running time.

(b) Consider a greedy algorithm that cuts the wood so that the maximun length of the resulting two pieces is always as small as possible. Show that this algorithm does not achieve minimal cost by giving a counter-example.

Problem 6-2. Do Exercise 24.1–8 on page 503 of CLR, 23.1–8 on page 567 of CLRS. [50 points]