CS 5114
Solutions to Midterm Exam
March 2, 2000

[30] 1. Rank the following functions by order of growth; that is, name the functions g_1, g_2, g_3, g_4 so that $g_1 = \omega(g_2)$, $g_2 = \omega(g_3)$, and $g_3 = \omega(g_4)$. (Note that the ranking is strict.)

$$\frac{5n^7+2.7n^3}{3.5n^4+17n} \qquad 2^{3\lg n \lg\lg n} \qquad \lg(n!) \qquad \frac{n^3}{(\lg n)^2}.$$

Prove your ranking.

First express each function asymptotically as a power of n:

$$\begin{aligned}
\frac{5n^7+2.7n^3}{3.5n^4+17n} &= \Theta(n^3) \\
2^{3\lg n \lg\lg n} &= n^{3\lg\lg n} \\
\lg(n!) &= \Theta(n \lg n) \\
&= \Theta\left(n^{1+(\lg\lg n)/(\lg n)}\right) \\
\frac{n^3}{(\lg n)^2} &= \Theta\left(n^{3-2(\lg\lg n)/(\lg n)}\right).
\end{aligned}$$

Notice that the asymptotic result for $\lg(n!)$ was obtained in the solutions for Homework 1.

It is clear that the ranking should be

$$\begin{aligned}
g_1(n) &= 2^{3\lg n \lg\lg n} \\
g_2(n) &= \frac{5n^7+2.7n^3}{3.5n^4+17n} \\
g_3(n) &= \frac{n^3}{(\lg n)^2} \\
g_4(n) &= \lg(n!).
\end{aligned}$$

The ranking is clear BECAUSE the previous asymptotic expressions are in similar forms (as powers of n).

If the rankings were just "guessed", then a proof of the rankings might go as follows:

- $$\begin{aligned}
\lim_{n\to\infty} \frac{g_1(n)}{g_2(n)} &= \lim_{n\to\infty} \frac{2^{3\lg n \lg\lg n}(3.5n^4+17n)}{5n^7+2.7n^3} \\
&= \lim_{n\to\infty} \frac{n^{3\lg\lg n}(3.5n^4+17n)}{5n^7+2.7n^3} \\
&= \lim_{n\to\infty} \frac{n^{3\lg\lg n-3}(3.5n^7+17n^4)}{5n^7+2.7n^3} \\
&= \lim_{n\to\infty} (7/10) n^{3\lg\lg n-3} \\
&= \infty,
\end{aligned}$$

because $3\lg\lg n - 3 > 0$ for sufficiently large n.

-
$$\begin{aligned}
\lim_{n\to\infty} \frac{g_2(n)}{g_3(n)} &= \lim_{n\to\infty} \frac{(5n^7 + 2.7n^3)(\lg n)^2}{(3.5n^4 + 17n)n^3} \\
&= \lim_{n\to\infty} \frac{(5n^7 + 2.7n^3)(\lg n)^2}{3.5n^7 + 17n^4} \\
&= \lim_{n\to\infty} (10/7)(\lg n)^2 \\
&= \infty,
\end{aligned}$$

because $(\lg n)^2 \to \infty$ as $n \to \infty$.

-
$$\begin{aligned}
\lim_{n\to\infty} \frac{g_3(n)}{g_4(n)} &= \lim_{n\to\infty} \frac{n^3}{(\lg n)^2 \lg(n!)} \\
&= \infty,
\end{aligned}$$

because $(\lg n)^2 \lg(n!) = o(n^2)$.

---

[30] 2. Give asymptotic upper and lower bounds for $T(n)$ in each of the following recurrences. Make your bounds as tight as possible and prove them.

A. $T(n) = 4T(n/3) + n^{1/2}$

B. $T(n) = 27T(n/13) + 3^n$

A. We apply the Master Theorem with $a = 4$, $b = 3$, and $f(n) = n^{1/2}$. Since $\log_b a = \log_3 4 > 1$, we get that case 1 of the Master Theorem applies, from which we obtain

$$T(n) = \Theta\left(n^{\log_3 4}\right).$$

B. We apply the Master Theorem with $a = 27$, $b = 13$, and $f(n) = 3^n$. Since $f(n)$ is asymptotically greater than any polynomial in n, including $n^{\log_b a}$, we get that case 3 of the Master Theorem applies, from which we obtain

$$\begin{aligned}
T(n) &= \Theta(f(n)) \\
&= \Theta(3^n).
\end{aligned}$$

---

[40] 3. Consider the following counting variation of the knapsack problem that counts the number of ways that some of the N items can fit *exactly* into a knapsack of size M.

KNAPSACK COUNTING PROBLEM

INSTANCE: N items $1, 2, \ldots, N$ with positive integer sizes $s_1, s_2, \ldots, s_N$; a positive integer knapsack size M.

SOLUTION: The count C of the number of sets of items $S \subset \{1, 2, \ldots, N\}$ such that $\sum_{i \in S} s_i = M$.

If we define a predicate f on sets of items by

$$f(S) \quad = \quad \begin{cases} 1 & if \text{if } \sum_{i \in S} s_i = M; \\ 0 & \text{otherwise,} \end{cases}$$

then the solution to the instance is $C = \sum_{S \subset \{1,2,\ldots,N\}} f(S)$.

EXAMPLE. Given the instance with $N = 5$, $M = 7$, and

$$\begin{aligned} s_1 &= 1 \\ s_2 &= 7 \\ s_3 &= 4 \\ s_4 &= 3 \\ s_5 &= 3, \end{aligned}$$

we get $C = 4$, since the following subsets of items have sizes summing to M:

$$\{s_1, s_4, s_5\} \qquad \{s_2\} \qquad \{s_3, s_4\} \qquad \{s_3, s_5\}.$$

A. Use the dynamic programming paradigm to develop an algorithm to return C for an instance of the KNAPSACK COUNTING PROBLEM[1]. Give pseudocode for your algorithm.

B. Analyze the time and space complexity of your algorithm.

C. Fill in the table of values for subproblems that result from executing your algorithm on the example above.

---

[1] If you wish, you may take the subproblems to consist of items $1, 2, \ldots, i$ and knapsack size j, where $1 \le i \le N$ and $0 \le j \le M$. The value to store for subproblem i, j is then $E[i, j]$, the count of subsets of $\{1, 2, \ldots, i\}$ that exactly fit in a knapsack of size j.

```
KCP(N; s_1, s_2, ···, s_N; M)
    ▷ Handle the base case, j = 0.
    for i ← 1 to N
        do E[i, 0] ← 1
    ▷ Handle the base case, i = 1.
    for j ← 0 to M
        do if s_i = j
              then E[i, 0] ← 1
              else E[i, 0] ← 0
    ▷ Handle the general case.
    for i ← 2 to N
        do for j ← 1 to M
              do if s_i > j
                    then E[i, j] ← E[i − 1, j]
                    else E[i, j] ← E[i − 1, j] + E[i − 1, j − s_i]
    return E[N, M]          ▷ Return C
```

Figure 1: Dynamic programming algorithm for the Knapsack Counting Problem.

A. Using the suggestion, we take the subproblems to consist of items $1, 2, \ldots, i$ and knapsack size j, where $1 \leq i \leq N$ and $0 \leq j \leq M$. The value to store for subproblem i, j is then $E[i, j]$, the count of subsets of $\{1, 2, \ldots, i\}$ that exactly fit in a knapsack of size j.

The base cases occur when $j = 0$ and $E[i, 0] = 1$ (the empty set); and when $i = 1$ and

$$E[1, j] \quad = \quad \begin{cases} 1 & \text{if } s_1 = j; \\ 0 & \text{otherwise.} \end{cases}$$

The general case occurs when $2 \leq i \leq N$ and $1 \leq j \leq M$. We have

$$E[i, j] \quad = \quad \begin{cases} E[i-1, j] & \text{if } s_i > j; \\ E[i-1, j] + E[i-1, j-s_i] & \text{if } s_i \leq j. \end{cases}$$

The pseudocode for the resulting dynamic programming algorithm KCP can be found in Figure 1.

B. For each of $N(M + 1)$ subproblems, the algorithm remembers one integer. Hence the space complexity is $\Theta(NM)$. The time to compute each $E[i, j]$ is $\Theta(1)$. Hence the time complexity is also $\Theta(NM)$.

C. Here is the table

i \ j	0	1	2	3	4	5	6	7
1	1	1	0	0	0	0	0	0
2	1	1	0	0	0	0	0	1
3	1	1	0	0	1	1	0	1
4	1	1	0	1	2	1	0	2
5	1	1	0	2	3	1	1	4