

3.5 Compare the following pairs of functions in terms of order of magnitude. In each case, say whether  $f(n) = O(g(n))$ ,  $f(n) = \Omega(g(n))$ , and/or  $f(n) = \Theta(g(n))$ .

	$f(n)$	$g(n)$
a.	$100n + \log n$	$n + (\log n)^2$
b.	$\log n$	$\log(n^2)$
c.	$\frac{n^2}{\log n}$	$n(\log n)^2$
d.	$(\log n)^{\log n}$	$\frac{n}{\log n}$
e.	$n^{1/2}$	$(\log n)^5$
f.	$n 2^n$	$3^n$

3.20 Prove that the asymptotic solution for the recurrence relation

$$T(n) = 2T(n - c) + k,$$

where both  $c$  and  $k$  are integer constants, is  $T(n) = O(d^n)$  for some constant  $d$ .

5.10 Solve the following **extended skyline problem**. Suppose that the buildings in the skyline have roofs. Each building is a rectangle with a triangular roof on top. (You can assume for simplicity that all the roofs have 45-degree angles with the buildings.) Again, all the buildings have a common horizon. Design an algorithm to draw the skyline in this case.

5.17 Solve the following variation of the knapsack problem: The assumptions are identical to those of Section 5.10, except that there is an unlimited supply of each item. In other words, the problem is to pack items of given sizes in a given-sized knapsack, but each item may appear many times.

5.20 Let  $x_1, x_2, \dots, x_n$  be a set of integers, and let  $S = \sum_{i=1}^n x_i$ . Design an algorithm to partition the set into two subsets of equal sum, or determine that it is impossible to do so. The algorithm should run in time  $O(nS)$ .

**مسئله:** یک درخت داده شده است که روی هر راس آن یک عدد صحیح نوشته شده است. می‌خواهیم یک مسیر روی درخت پیدا کنیم که جمع اعدادی که روی راس‌های آن نوشته شده بیشینه باشد. زمان اجرای الگوریتم شما باید  $O(n)$  باشد.

**مسئله:** تمرین برنامه‌نویسی: دنباله‌ای از اعداد صحیح داده شده است. زیر دنباله‌ای متوالی از آن را بیابید که جمع اعداد آن بیشینه باشد. زمان اجرای الگوریتم شما باید  $O(n)$  باشد.  
ورودی: در یک خط  $n$  تعداد اعداد و در سطر بعدی اعداد به ترتیب با یک فاصله نوشته شده اند.  
خروجی: بیشینه جمع زیردنباله‌های متوالی