Chris Grimes
Algorithms
Hw #5
1. Let S = {a, b, c, d, e, f, g} be a collection of items with weight-benefit values as follows: a(3, 12),

b(6, 12), c(6, 9), d(1, 5), e(2, 5), f(10, 10), g(3, 9). For example, item a weighs 3 lbs and is worth a total

of $12. Assume you have a knapsack that can hold a total of 11 lbs.

(a) (10 points) What is an optimal solution to the fractional knapsack problem for S? Show your

work.

With the values above: a(3, 12), b(6, 12), c(6, 9), d(1, 5), e(2, 5), f(10, 10), g(3, 9)

the items benefit per weight values are: a(4), b(2), c(1.5), d(5), e(2.5), f(1), g(3)

and a total knapsack weight of 11lbs:

first take all of item d which is 1 lb, knapsack now has{1 lb of d} total weight is 1 lb total value is $5

then take all of a which is 3 lbs, knapsack now has{1 lb of d, 3 lbs of a} total weight is 4 lbs total value is $17

next take all of g which is 3 lbs, knapsack now has{1 lb of d, 3 lbs of a, 3 lbs of g} total weight is 7 lbs total value is $26

next take all of e which is 2 lbs, knapsack now has{1 lb of d, 3 lbs of a, 3 lbs of g, 2 lbs of e} total weight is 9 lbs total value is $31

finally take 2 lbs of b, knapsack now has{1 lb of d, 3 lbs of a, 3 lbs of g, 2 lbs of e, 2 lbs of b} total weight is 11 lbs total value is $35

(b) (10 points) What is an optimal solution to the 0-1 knapsack problem for S? Show your work.

With the values above: a(3, 12), b(6, 12), c(6, 9), d(1, 5), e(2, 5), f(10, 10), g(3, 9)

the items benefit per weight values are: a(4), b(2), c(1.5), d(5), e(2.5), f(1), g(3)

and a total knapsack weight of 11lbs:

first take all of item d which is 1 lb, knapsack now has{1 lb of d} total weight is 1 lb total value is $5

then take all of a which is 3 lbs, knapsack now has{1 lb of d, 3 lbs of a} total weight is 4 lbs total value is $17

next take all of g which is 3 lbs, knapsack now has{1 lb of d, 3 lbs of a, 3 lbs of g} total weight is 7 lbs total value is $26

finally take all of e which is 2 lbs, knapsack now has{1 lb of d, 3 lbs of a, 3 lbs of g, 2 lbs of e} total weight is 9 lbs total value is $31

This is the most we can fit in our knapsack given that we take all or none of a given item.

Chris Grimes
Algorithms
Hw #5
2. (5 points) Suppose we are given a set of tasks specified by pairs of start times and finish times

as T = {(5, 6),(9, 11)(3, 7),(1, 2),(10, 12),(6, 8),(1, 3),(7, 9),(1, 4),(11, 14),(2, 5),(4, 9),(7, 10)}. Solve

the task scheduling problem for these tasks.

Beginning with no tasks currently scheduled:

first we'll order our list by start times:

T={(1, 2), (1, 3), (1, 4), (2, 5), (3, 7), (4, 9), (5, 6), (6, 8), (7, 9), (7, 10), (9, 11), (10, 12), (11, 14)}

we"ll assign our first task (1,2)

	1:00	2:00	3:00	4:00	5:00	6:00	7:00	8:00	9:00	10:00	11:00	12:00
Machine 7												
Machine 6												
Machine 5												
Machine 4												
Machine 4												
Machine 3												
Machine 2												
Machine 1	(1,2)											

next we'll assign the next task in T (1,3)

	1:00	2:00	3:00	4:00	5:00	6:00	7:00	8:00	9:00	10:00	11:00	12:00
Machine 7												
Machine 6												
Machine 5												
Machine 4												
Machine 4												
Machine 3												
Machine 2	(1,3)	(1,3)										
Machine 1	(1,2)											

next we'll assign the next task in T (1,4)

	1:00	2:00	3:00	4:00	5:00	6:00	7:00	8:00	9:00	10:00	11:00	12:00
Machine 7												
Machine 6												
Machine 5												
Machine 4												
Machine 4												
Machine 3	(1,4)	(1,4)	(1,4)									
Machine 2	(1,3)	(1,3)										
Machine 1	(1,2)											

Chris Grimes
Algorithms
Hw #5
next we'll assign the next task in T (2, 5)

	1:00	2:00	3:00	4:00	5:00	6:00	7:00	8:00	9:00	10:00	11:00	12:00
Machine 7												
Machine 6												
Machine 5												
Machine 4												
Machine 4												
Machine 3	(1,4)	(1,4)	(1,4)									
Machine 2	(1,3)	(1,3)										
Machine 1	(1,2)	(2, 5)	(2, 5)	(2, 5)								

next we'll assign the next task in T (3, 7)

	1:00	2:00	3:00	4:00	5:00	6:00	7:00	8:00	9:00	10:00	11:00	12:00
Machine 7												
Machine 6												
Machine 5												
Machine 4												
Machine 4												
Machine 3	(1,4)	(1,4)	(1,4)									
Machine 2	(1,3)	(1,3)	(3, 7)	(3, 7)	(3, 7)	(3, 7)						
Machine 1	(1,2)	(2, 5)	(2, 5)	(2, 5)								

next we'll assign the next task in T (4, 9)

	1:00	2:00	3:00	4:00	5:00	6:00	7:00	8:00	9:00	10:00	11:00	12:00
Machine 7												
Machine 6												
Machine 5												
Machine 4												
Machine 4												
Machine 3	(1,4)	(1,4)	(1,4)	(4, 9)	(4, 9)	(4, 9)	(4, 9)	(4, 9)				
Machine 2	(1,3)	(1,3)	(3, 7)	(3, 7)	(3, 7)	(3, 7)						
Machine 1	(1,2)	(2, 5)	(2, 5)	(2, 5)								

Chris Grimes
Algorithms
Hw #5
next we'll assign the next task in T (5, 6)

	1:00	2:00	3:00	4:00	5:00	6:00	7:00	8:00	9:00	10:00	11:00	12:00
Machine 7												
Machine 6												
Machine 5												
Machine 4												
Machine 4												
Machine 3	(1,4)	(1,4)	(1,4)	(4, 9)	(4, 9)	(4, 9)	(4, 9)	(4, 9)				
Machine 2	(1,3)	(1,3)	(3, 7)	(3, 7)	(3, 7)	(3, 7)						
Machine 1	(1,2)	(2, 5)	(2, 5)	(2, 5)	(5, 6)							

next we'll assign the next task in T (6, 8)

	1:00	2:00	3:00	4:00	5:00	6:00	7:00	8:00	9:00	10:00	11:00	12:00
Machine 7												
Machine 6												
Machine 5												
Machine 4												
Machine 4												
Machine 3	(1,4)	(1,4)	(1,4)	(4, 9)	(4, 9)	(4, 9)	(4, 9)	(4, 9)				
Machine 2	(1,3)	(1,3)	(3, 7)	(3, 7)	(3, 7)	(3, 7)						
Machine 1	(1,2)	(2, 5)	(2, 5)	(2, 5)	(5, 6)	(6, 8)	(6, 8)					

next we'll assign the next task in T (7, 9)

	1:00	2:00	3:00	4:00	5:00	6:00	7:00	8:00	9:00	10:00	11:00	12:00
Machine 7												
Machine 6												
Machine 5												
Machine 4												
Machine 4												
Machine 3	(1,4)	(1,4)	(1,4)	(4, 9)	(4, 9)	(4, 9)	(4, 9)	(4, 9)				
Machine 2	(1,3)	(1,3)	(3, 7)	(3, 7)	(3, 7)	(3, 7)	(7, 9)	(7, 9)				
Machine 1	(1,2)	(2, 5)	(2, 5)	(2, 5)	(5, 6)	(6, 8)	(6, 8)					

Chris Grimes
Algorithms
Hw #5
next we'll assign the next task in T (7, 10)

	1:00	2:00	3:00	4:00	5:00	6:00	7:00	8:00	9:00	10:00	11:00	12:00
Machine 7												
Machine 6												
Machine 5												
Machine 4												
Machine 4							(7, 10)	(7, 10)	(7, 10)			
Machine 3	(1,4)	(1,4)	(1,4)	(4, 9)	(4, 9)	(4, 9)	(4, 9)	(4, 9)				
Machine 2	(1,3)	(1,3)	(3, 7)	(3, 7)	(3, 7)	(3, 7)	(7, 9)	(7, 9)				
Machine 1	(1,2)	(2, 5)	(2, 5)	(2, 5)	(5, 6)	(6, 8)	(6, 8)					

next we'll assign the next task in T (9, 11)

	1:00	2:00	3:00	4:00	5:00	6:00	7:00	8:00	9:00	10:00	11:00	12:00
Machine 7												
Machine 6												
Machine 5												
Machine 4												
Machine 4							(7, 10)	(7, 10)	(7, 10)			
Machine 3	(1,4)	(1,4)	(1,4)	(4, 9)	(4, 9)	(4, 9)	(4, 9)	(4, 9)				
Machine 2	(1,3)	(1,3)	(3, 7)	(3, 7)	(3, 7)	(3, 7)	(7, 9)	(7, 9)				
Machine 1	(1,2)	(2, 5)	(2, 5)	(2, 5)	(5, 6)	(6, 8)	(6, 8)		(9, 11)	(9, 11)		

next we'll assign the next task in T (10, 12)

	1:00	2:00	3:00	4:00	5:00	6:00	7:00	8:00	9:00	10:00	11:00	12:00
Machine 7												
Machine 6												
Machine 5												
Machine 4												
Machine 4							(7, 10)	(7, 10)	(7, 10)			
Machine 3	(1,4)	(1,4)	(1,4)	(4, 9)	(4, 9)	(4, 9)	(4, 9)	(4, 9)				
Machine 2	(1,3)	(1,3)	(3, 7)	(3, 7)	(3, 7)	(3, 7)	(7, 9)	(7, 9)		(10, 12)	(10, 12)	
Machine 1	(1,2)	(2, 5)	(2, 5)	(2, 5)	(5, 6)	(6, 8)	(6, 8)		(9, 11)	(9, 11)		

Chris Grimes
Algorithms
Hw #5
next we'll assign the next task in T (11, 14)

	1:00	2:00	3:00	4:00	5:00	6:00	7:00	8:00	9:00	10:00	11:00	12:00	13:00
Machine 7													
Machine 6													
Machine 5													
Machine 4													
Machine 4							(7, 10)	(7, 10)	(7, 10)				
Machine 3	(1,4)	(1,4)	(1,4)	(4, 9)	(4, 9)	(4, 9)	(4, 9)	(4, 9)			(11, 14)	(11, 14)	(11, 14)
Machine 2	(1,3)	(1,3)	(3, 7)	(3, 7)	(3, 7)	(3, 7)	(7, 9)	(7, 9)		(10, 12)	(10, 12)		
Machine 1	(1,2)	(2, 5)	(2, 5)	(2, 5)	(5, 6)	(6, 8)	(6, 8)		(9, 11)	(9, 11)			

Chris Grimes
Algorithms
Hw #5

3. Consider the single machine scheduling problem where we are given a set T of tasks specified by their start times and finish times, as in the task scheduling problem, except now we have only one machine and we wish to maximize the number of tasks that this single machine performs.

(a) (7 points) Design a greedy algorithm for this single machine scheduling problem. What is the running time of your algorithm?

```
Algorithm schedule(set of tasks T with n tasks):
    input: a set of tasks T with n tasks
    output: an array with our scheduled tasks
    sort(T based on finish times in ascending order)
    int j=0
    for(int i=0; i<n-1;)
        if(i==0 && j==0)
            X[i]=T[j]
            ++j
            ++i
        else if(T[j]->beginTime() >= X[i-1]->endTime())
            X[i]=T[j]
            ++j
            ++i
        else
            ++j
    return X
```

O(n log(n)) do to using sort

Chris Grimes
Algorithms
Hw #5
(b) (3 points) Prove the correctness of your algorithm.

Given T = {(5, 6),(9, 11)(3, 7),(1, 2),(10, 12),(6, 8),(1, 3),(7, 9),(1, 4),(11, 14),(2, 5),(4, 9),(7, 10)}

Algorithm schedule(set of tasks T with n tasks):

```
    input: a set of tasks T with n tasks

    output: an array with our scheduled tasks

    sort(T based on finish times in ascending order)

    T={(1, 2), (1, 3), (1, 4), (2, 5), (5, 6), (3, 7), (6, 8), (4, 9),( 7, 9), (7, 10), (9, 11), (10, 12), (11, 14)}

    int j=0

    for(int i=0; i<n-1;)

        if(i==0 && j==0)

            X[i]=T[j]

            ++j

            ++i

        else if(T[j]->beginTime() >= X[i-1]->endTime())

            X[i]=T[j]

            ++j

            ++i

        else

            ++j

    return X
```

Output: X={(1,2), (2, 5), (5, 6), (6, 8), (9, 11), (11, 14)}

Chris Grimes
Algorithms
Hw #5
4. (10 points) For each of the following recurrence equations which describe the running time T(n) of a recursive algorithm, use the master method to express the asymptotic complexity (assuming that T(n) = c for n < d, for constants c > 0 and d ≥ 1).

(a) T(n) = 2T(n/2) + log n

$\log_b a$=1

f(n)=log(n)

so, case 1 says T(n) is big theta of n

(b) T(n) = 8T(n/2) + n^2

$\log_b a$=3

f(n)=n^2

so, case 1 says T(n) is big theta of n^3

(c) T(n) = 7T(n/3) + n

$\log_b a$= 1.7712437492

f(n)=n

so, case 1 says T(n) is big theta of $n^{1.7712437492}$

(d) T(n) = 4T(n/2) + n^2

$\log_b a$= 2

f(n)=n^2

so, case 2 says T(n) is big theta of n^2 log(n) n^2

(e) T(n) = 3T(n/2) + n^2

$\log_b a$= 1.5849625007

f(n)= n^2

so, case 3 says T(n) is big theta of n^2

Chris Grimes
Algorithms
Hw #5
5. (10 points) What is the best way to multiply a chain of matrices with dimensions that are 10 × 5, 5 × 2, 2 × 20, 20 × 12, 12 × 4, and 4 × 60? Show your work (including two tables - one indicating the index k which gives the final multiply index for each subproblem, and the second table which indicates the optimal number of multiplications for each subproblem).

A_0 A_1 A_2 A_3 A_4 A_5

Using the set A={(10 × 5), (5 × 2), (2 × 20), (20 × 12), (12 × 4), (4 × 60)}

N	0	1	2	3	4	5
0	0	100	500	820	756	2356
1		0	200	600	616	1656
2			0	480	576	1056
3				0	960	5760
4					0	2880
5						0

k	0	1	2	3	4	5
0		0	1	1	1	1
1			1	1	1	1
2				2	3	4
3					3	4
4						4
5						

A0*A1 N[0][1]=0+0+10*5*2=100 A1*A2 N[1][2]=0+0+5*2*20=200

A2*A3 N[2][3]=0+0+2*20*12=480 A3*A4 N[3][4]=0+0+20*12*4=960

A4*A5 N[4][5]=0+0+12*4*60=2880

(A0)(A1*A2) = 0+200+10*5*20=1200 (A0*A1)(A2)=100+0+10*2*20=500

(A1)(A2*A3)=0+480+5*2*12=600 (A1*A2)(A3)=200+0+5*20*12=1400

(A2)(A3*A4)=0+960+2*20*4=1120 (A2*A3)(A4)=480+0+2*12*4=576

(A3)(A4*A5)=0+2880+20*12*60=17280 (A3*A4)(A5)=960+0+20*4*60=5760

Chris Grimes
Algorithms
Hw #5
(A0)(A1*A2*A3)=0+600+10*5*12=1200 (A0*A1)(A2*A3)=100+480+10*2*12=820

(A0*A1*A2)(A3)=500+0+10*20*12=2900

(A1)(A2*A3*A4)=0+576+5*2*4=616 (A1*A2)(A3*A4)=200+960+5*20*4=1560

(A1*A2*A3)(A4)=600+0+5*12*4=840

(A2)(A3*A4*A5)=0+5760+2*20*60=8160 (A2*A3)(A4*A5)=480+0+2*12*60=1920

(A2*A3*A4)(A5)=576+0+2*4*60=1056

(A0)(A1*A2*A3*A4)=0+616+10*5*4=816 (A0*A1)(A2*A3*A4)=100+576+10*2*4=756

(A0*A1*A2)(A3*A4)=500+960+10*20*4=2260 (A0*A1*A2*A3)(A4)=820+0+10*12*4=1300

(A1)(A2*A3*A4*A5)=0+1056+5*2*60=1656 (A1*A2)(A3*A4*A5)=200+5760+5*20*60=11960

(A1*A2*A3)(A4*A5)=600+2880+5*12*60=7080 (A1*A2*A3*A4)(A5)=616+0+5*4*60=1816

(A0)(A1*A2*A3*A4*A5)=0+1656+10*5*60=4656

(A0*A1)(A2*A3*A4*A5)=100+1056+10*2*60=2356

(A0*A1*A2)(A3*A4*A5)=500+5760+10*20*60=18260

(A0*A1*A2*A3)(A4*A5)=820+2880+10*12*60=10900

(A0*A1*A2*A3*A4)(A5)=756+0+10*4*60=3156

optimal order: [(A0*A1)]*[(A2*A3*A4)*(A5)]