Institut for Matematik og Datalogi
Syddansk Universitet

November 27, 2008
JFB

Introduction to Computer Science
E08 – Week 12

Announcement

IMADA JULEFROKOST 12/12 kl. 14.00 i U49!!! KOM OG VÆR MED TIL EN MASSE HYGGE, SJOV OG SPAS! DER VIL BLIVE SERVERET GLØGG OG EN TRADITIONEL JULEFROKOSTMENU.

DET ER MULIGT AT KØBE (JULE)ØL, VAND OG SNAPS.

EFTER FROKOSTEN FORTSÆTTER FESTLIGHEDERNE TIL FAKULTETETS JULEFEST PÅ CAMPUSTORV.

PRIS: Kr. 140.-

TILMELDING OG BETALING PÅ SEKRETARIATET SENEST FREDAG 5/12 KL. 12.00

Repeat announcement

Imada holder pizzamøde for alle studerende mandag 1/12 kl. 16.15 i U49. Mødet vil indeholde generel information om kandidat- og bachelorstudiet, samt orientering om planlagte valgfri kurser i næste semester. Til slut vil der være gratis pizza, øl og sodavand til de fremmødte.

There will be a "pizza-meeting" for all students of Imada on Monday, December 1 at 16.15 in room U49. At the meeting Imada will give general information on the bachelor and candidate studies, and specific information on the elective courses planned for the next semester. The meeting will end with a free pizza, beer, and soft drink session.

Lecture, November 24

We covered the first five sections of chapter 7 in the textbook.

Lecture, November 26

We covered section 7.6 in the textbook and introduced on-line algorithms.

Lecture, December 1

We will begin on chapter 11 in the textbook.

Lecture, December 8

We will continue with chapter 11 in the textbook.

Discussion section: week 50

Discuss the following problems in groups of 3 to 4. (Think about these problems before coming to discussion section.)

1. The List Scheduling algorithm (LS) places each new job on the machine with lowest load (currently). We showed in class that the algorithm has a competitve ratio of $2 - \frac{1}{m}$, where m is the number of machines. (Note the competitive ratio is the worst case ratio, over all possible input sequences, of the value the on-line algorithm achieves on the input sequence to the value the optimal off-line algorithm achieves on the same input sequence.) In order to show that the competitive ratio was this high, we showed that if the algorithm gets many small jobs ($m(m-1)$ of length 1) followed by one large job (a job of length m), LS will pack them so each machine has the same number of the small ones, so the last job will be placed on some machine that already has a large load. One might hope that holding a single machine free for such large jobs could help.

   Consider the following algorithm: Machine M_1 is kept mostly free for large jobs. The other machines are $M_2, M_3, ..., M_m$. Suppose you are trying to obtain a competitive ratio of C. When handling a job J, place it on the least loaded of machines $M_2, M_3, ..., M_m$ if placing it there does not bring the competitive ratio above C (if J was the last job). Otherwise, place it on machine M_1. Show that this strategy cannot lead to a better competitive ratio than $2 - \frac{1}{m}$.

2. If the number of machines is $m = 2$, then LS is 3/2-competitive and is optimal, i.e., no on-line algorithm has a better competitive ratio.

   Assume now, that one of the two machines is twice as fast as the other. Consider a variant of LS, LS2, which places a job J on the machine where it will finish first. This algorithm is 3/2-competitive. Show that this is optimal. (You can do it using a sequence with two jobs.)

   There are also other optimal algorithms for this problem. What other algorithm (very simple) also has competitive ratio 3/2?

3. Consider the dual bin-packing problem, which is the problem of putting (as many as possible) items into n bins, which all have size 1. First-Fit (FF), is the algorithm which puts an item (which arrived on-line) in the first bin in which it fits (no bin may be filled to more than 1). An *accommodation sequence* is a sequence which can fit in the n bins. Show that with an accommodating sequence, FF has a competitive ratio between $\frac{1}{2}$ and $\frac{4}{5}$, i.e., with such a sequence, FF packs at least half of the items in the bins, and there exists such a sequence where FF only packs $\frac{4}{5}$ of the items. You can do this using only items of size $\frac{1}{2}$ and $\frac{1}{3}$.

   Can you come further down than 4/5 by using items of size $\frac{1}{5}$, too?

4. In the classical bin-packing problem, you are not given a limit on how many bins there are and must pack all items. The goal is to use as few bins as possible. Show that FF has a competitive ratio between $\frac{5}{3}$ and 2. (Hint: you can show that it is at least $\frac{5}{3}$ using a sequence consisting of items of sizes $\frac{1}{7} + \frac{1}{2000}, \frac{1}{3} + \frac{1}{2000}, \frac{1}{2} + \frac{1}{2000}$.)

   Improve the $\frac{5}{3}$ to about 1.69 by using items of sizes $\frac{1}{43} + \frac{1}{8000}, \frac{1}{7} + \frac{1}{8000}, \frac{1}{3} + \frac{1}{8000}, \frac{1}{2} + \frac{1}{8000}$).

5. Show that FWF (the Flush-When-Full algorithm for paging) has a competitive ratio of at least k, where k is the cache size (show that there exist families of sequences, one sequence for each natural number, where asymptotically FWF faults k times as often as the optimal offline algorithm).

6. Show that FWF is never better than LRU.

Assignment due 14:15, December 18

Late assignments will not be accepted. Working together is not allowed. (You may write this either in English or Danish, but write clearly if you do it by hand.) Show your work.

1. The List Scheduling algorithm (LS) places each new job on the machine with lowest load (currently). We showed in class that the algorithm has a competitve ratio of $2 - \frac{1}{m}$, where m is the number of machines. (Note the competitive ratio is the worst case ratio, over all possible input sequences, of the value the on-line algorithm achieves on the input sequence to the value the optimal off-line algorithm achieves on the same input sequence. We are minimizing makespan, the time when the last job finishes.) In order to show that the competitive ratio was this high, we showed that if the algorithm gets many small jobs ($m(m-1)$ of length 1) followed by one large job (a job of length m), LS will pack them so each machine has the same number of the small ones, so the last job will be placed on some machine that already has a large load. One might hope that holding a single machine partially free for such large jobs could help.

   Consider the following algorithm: Machine M_1 is kept partially free for large jobs. The other machines are $M_2, M_3, ..., M_m$. When handling a job J, place it on the least loaded machine, unless that machine is M_1. If that machine is M_1, and M_1 would still be no more than half as loaded as the most heavily loaded of the other machines, place J on M_1. Otherwise, if the least heavily loaded machine is M_1, if J has length at least $3/4$ the load on the most heavily loaded machine, place J on M_1. If neither of these cases holds, place J on the least heavily loaded machine among $M_2, M_3, ..., M_m$. Show that this strategy cannot lead to a better competitive ratio than $2 - \frac{1}{m}$, for $m = 5$ machines (i.e., give a sequence for which this algorithm will have a machine with load $9/5$ times the maximum load the optimal off-line algorithm can achieve, when there are five machines).

2. Do problem 22 on page 567.