Lecture No. 2, 9/19. Topics

COS 521
Fall 2002
Tarjan

Open problem #4: Consider generating a sequence of symbols by beginning with an empty stack, executing a sequence of the following operations, and ending with an empty stack:

> push a: push symbol a onto the stack
> top: write the top symbol on the stack (*without* popping it!)
> pop: pop the top symbol off of the stack
>
> Given an n-symbol sequence, we want a shortest sequence of stack operations that generate it. There is a straightforward $O(n^3)$-time dynamic programming algorithm. It is likely that fast matrix multiplication can be applied. The open problem is to do better, e.g. $O(n^2)$.

Open problem #5: (stable partition): Given an array of n items and a predicate on the items, the goal is to permute the items in the array so that all "true" items precede all "false" ones, and the relative order of true items, and the relative order of false items, do not change. Can this be done in O(n) time and constant words of extra space?

Amortization: averaging the cost of operations over a (worst-case) sequence of operations

Amortized analysis via a potential function

R. Tarjan, "Amortized computational complexity," *SIAM J. Alg. Disc. Math.* 6 (1985), 306-318.

Amortized analysis of splay trees

Amortized cost of splaying

D. Sleator and R. Tarjan, Self-adjusting binary search trees," *J. Assoc. Comput. Mach.* 32 (1985), 652-686.