

## MAC 338 - Análise de Algoritmos

*Departamento de Ciência da Computação*

Primeiro semestre de 2013

### Lista 6

1. (**Exercícios 18.1-2 do CLR**) Mostre que se uma operação **Decrementa** for incluída nas operações de manipulação de um contador binário com  $k$  bits,  $n$  operações poderiam custar tempo  $\Theta(nk)$ .
2. Suponha que desejamos não apenas incrementar um contador mas também algumas vezes reinicializá-lo com zero. Mostre como implementar um contador com um vetor binário de maneira que qualquer seqüência de  $n$  operações **incrementa1** e **zera\_contador** consuma tempo  $O(n)$ , desde que o contador esteja inicialmente com zero. (**Dica:** Mantenha um apontador para o 1 mais significativo do contador.)
3. Uma seqüência de operações sobre uma pilha é executada numa pilha cujo tamanho nunca excede  $k$ . Depois de cada  $k$  operações, uma cópia da pilha toda é feita para propósito de *back-up*. Mostre que o custo de  $n$  operações sobre a pilha, incluindo a operação de cópia para *back-up*, é  $O(n)$ , atribuindo valores adequados de créditos a cada operação.
4. Considere a implementação de lista ligada para representar conjuntos disjuntos. Sugira uma mudança simples da rotina UNION que não necessite do apontador **fim** para o último da lista de cada conjunto. Sua sugestão deve ser tal que, independente de estarmos ou não usando a heurística dos tamanhos (anexe no final a lista menor), o consumo assintótico de tempo de pior caso deve se manter igual.
5. Mostre que  $\lg(\lg^* n) = O(\lg^*(\lg n))$ .
6. Considere a implementação do union-find por árvores enraizadas. Escreva uma versão não recursiva do FINDSET com compressão de caminhos.
7. Considere a implementação do union-find por árvores enraizadas com compressão de caminhos e heurística dos ranks (a árvore de menor rank é pendurada na de maior rank no union). Considere uma seqüência qualquer (válida) de  $m$  operações MAKESET, FINDSET e LINK em que todas as operações LINK aparecem antes das operações FINDSET. Mostre que tal seqüência consome, no pior caso, tempo  $O(m)$ . O que acontece com o tempo consumido por uma seqüência deste tipo se apenas compressão de caminhos estiver implementada?