

MAC323/BCC TERCEIRO EXERCÍCIO-PROGRAMA

CONTAGEM DE POLIMINÓS

Y. KOHAYAKAWA

Data de entrega. Você deve entregar este EP até as 18:00 do dia 4/6/2001, na secretaria do Departamento de Ciência da Computação, sala 256, bloco A.

1. INTRODUÇÃO

Neste exercício-programa você investigará, através de um exemplo, o uso de estruturas de dados na implementação de algoritmos eficientes.

2. O PROBLEMA A SER RESOLVIDO

O programa que você escreverá neste exercício deve determinar o número de poliminós com n células. *Poliminós* generalizam dominós (que são poliminós com $n = 2$ células). Você já conhece os *pentaminós*, do [Exercício-Programa 2](#) de MAC122 (2o. semestre de 2000). Poliminós também são conhecidos como *animais*.¹

Formalmente, nossos poliminós são formados por quadrados de lado unitário (as células). As células em um poliminó devem formar uma configuração *conexa*, isto é, entre quaisquer duas células c e c' de nosso poliminó, deve haver uma seqüência de células $c_0, \dots, c_\ell$ com $c_0 = c$ e $c_\ell = c'$ e tal que c_{i-1} e c_i têm uma aresta comum para todo $1 \leq i \leq \ell$. Você encontra na Figura 1 um belo poliminó (ou um belo animal se você preferir).

Faremos a contagem de nossos poliminós “a menos de translação”. Isto é, identificamos dois poliminós X e Y se X pode ser obtido de Y por uma translação. Note que não permitimos rotação ou reflexão. De acordo com esta convenção, há dois dominós e há $2 + 4 = 6$ triminós.

Seja a_n o número de poliminós com n células. Os valores conhecidos de a_n são dados na Tabela 1 [veja I. Jensen, *Enumerations of lattice animals and trees*, a aparecer no Journal of Statistical Physics, disponível [arXiv:cond-mat/0007239](#)].

2.1. Programas de exemplo. É instrutivo estudar o programa escrito por Knuth, chamado [polynum](#), baseado no algoritmo de Jensen descrito em seu [artigo acima](#). A codificação de Knuth é bastante sofisticada, e assim um entendimento completo de seu programa pode estar além do escopo desta disciplina. Por outro lado, você pode aprender várias coisas interessantes estudando este programa e, portanto, recomendo fortemente que você dedique um certo tempo a ele. O programa [polynum](#) deve ser usado em conjunto com [polyslave](#).

Date: Versão de 9 de maio de 2001.

¹Você pode, portanto, dizer que este EP é o *EP animal*.

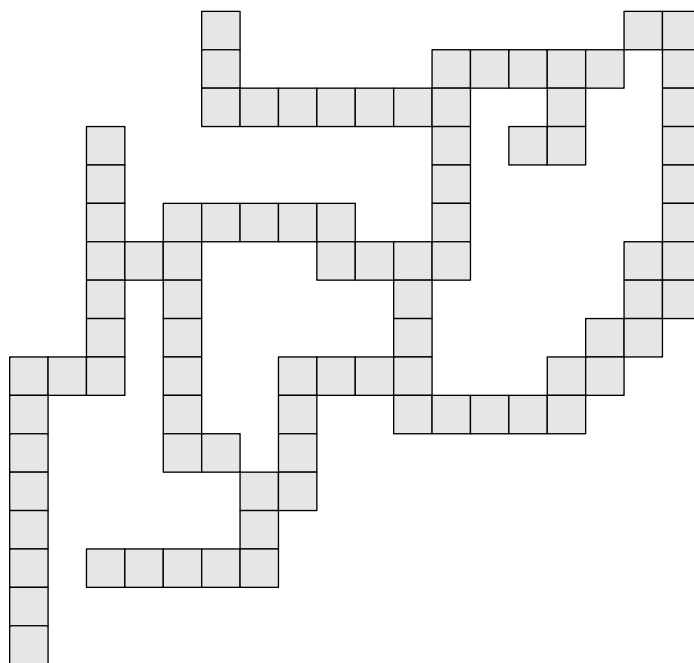


FIGURA 1. Um belo animal

Estudando cuidadosamente a estratégia de **polynum** e **polyslave**, você deverá ser capaz de escrever um programa que determina o número de animais a_n para valores de n moderadamente grandes.

2.1.1. *Recomendações.* Sugiro que você comece escrevendo um programa simples, para ver até qual valor de n você consegue chegar com técnicas elementares. Procure entender as partes mais simples de **polynum** e **polyslave** para ver se você consegue incorporar idéias do método de Jensen e Knuth. Você ainda pode achar interessante ver **um programa que Knuth escreveu** antes de conhecer o trabalho de Jensen.

Você pode entregar até duas versões deste EP: uma versão elementar e a melhor versão que você conseguir escrever.

3. OBSERVAÇÕES

1. *Este EP pode ser feito em duplas.*
2. Seja cuidadoso com sua programação (correção, documentação, apresentação, clareza do código, etc), dando especial atenção a suas estruturas de dados. A correção será feita levando isso em conta.
3. Comparem entre vocês o desempenho de seus programas.
4. Entregue o seu EP seguindo os moldes usuais: listagem impressa, cabeçalho claro, disquete com os arquivos (coloque um **Makefile**) e eventuais arquivos de teste.

Observação final. Enviem dúvidas para a lista de discussão da disciplina. Eventualmente, ajustes no enunciado ocorrerão ao discutirmos este EP na lista.

TABELA 1. Número de poliminós a_n

n	a_n
1	1
2	2
3	6
4	19
5	63
6	216
7	760
8	2725
9	9910
10	36446
11	135268
12	505861
13	1903890
14	7204874
15	27394666
16	104592937
17	400795844
18	1540820542
19	5940738676
20	22964779660
21	88983512783
22	345532572678
23	1344372335524
24	5239988770268
25	20457802016011
26	79992676367108
27	313224032098244
28	1228088671826973
29	4820975409710116
30	18946775782611174
31	74541651404935148
32	293560133910477776
33	1157186142148293638
34	4565553929115769162
35	18027932215016128134
36	71242712815411950635
37	281746550485032531911
38	1115021869572604692100
39	4415695134978868448596
40	17498111172838312982542
41	69381900728932743048483
42	275265412856343074274146
43	1092687308874612006972082
44	4339784013643393384603906
45	17244800728846724289191074
46	68557762666345165410168738