

Kleine Einführung in studentische Algorithmik-Programmierwettbewerbe anhand des ACM International Collegiate Programming Contest

Andreas Ruprecht, Bernhard Heinloth

Konferenz der Informatikfachschaften 41.5
an der Friedrich-Alexander-Universität Erlangen-Nürnberg

31st October 2013

Inhaltsverzeichnis

1 Übersicht

2 Wettbewerbe

Beispiele

ICPC

3 Allgemeines

Aufgaben

Abgaben

Scoreboard

4 Tipps

Aufgabenauswahl

Strategien bei Fehler

5 Üben

Beispiellösung

Übungsumgebung

6 Fragen

Bekannte Wettbewerbe

Von Firmen (jährlich)

dient natürlich primär der Suche von Nachwuchstalenten für das eigene Unternehmen

Google Code Jam

<http://code.google.com/codejam/>



Facebook Hacker Cup

<http://www.facebook.com/hackercup>



Bekannte Wettbewerbe

Von Firmen (jährlich)

dient natürlich primär der Suche von Nachwuchstalenten für das eigene Unternehmen

Google Code Jam

<http://code.google.com/codejam/>



Facebook Hacker Cup

<http://www.facebook.com/hackercup>



Online (Auswahl)

Codeforces

<http://codeforces.com/>



TopCoder Algorithms

<http://www.topcoder.com/>



Der ACM International Collegiate Programming Contest

„der älteste, größte und prestigeträchtigste Programmierwettbewerb der Welt“

Ausgangslage: Dein Team

- 3 Studierende (maximal im 10. Semester)
- ein Computer mit Standardumgebung
- ca. 10 Aufgaben
- 5 Stunden Zeit
- Verwendung der vorgegebenen Standardprogrammiersprachen (C, C++, Java)
- es darf ein Cheatsheet mit ≤ 25 Seiten mitgebracht werden
- kein Webzugriff (Leben ohne Google!)
- keine eigenen Daten (auch keine vorgefertigten Makefiles, Header, ...)

vgl. <http://icpc.baylor.edu/download/worldfinals/pdf/Factsheet.pdf>

Local Contests

- gibt es an manchen Universitäten
- um die Kandidaten für den Regional Contest zu bestimmen
- pro Uni kommen i.d.R. bis zu 3 Teams weiter
- weltweit mehr als 300.000 Teilnehmer



Local Contests

- gibt es an manchen Universitäten
- um die Kandidaten für den Regional Contest zu bestimmen
- pro Uni kommen i.d.R. bis zu 3 Teams weiter
- weltweit mehr als 300.000 Teilnehmer

In Erlangen

- gibt es im Wintersemester einen Fun-Contest
- im Sommer den Auswahlcontest für den NWERC
- meist 20+ Teams aus der FAU
- und ca. 40 Teams aus anderen Unis
- Teilnehmer aus allen Semestern (auch viele Erstis!)
- Lockmittel: kostenlos Pizzabrötchen, Kaffee & Süßkram

mehr Infos dazu auf <https://icpc.cs.fau.de/>

Regional Contests

- mehr als 30.000 Studierende
- von über 2.300 Universitäten
- aus über 90 Ländern



Regional Contests

- mehr als 30.000 Studierende
- von über 2.300 Universitäten
- aus über 90 Ländern

Für deutsche Unis

NWERC (Northwestern European Regional Contest)



vgl <http://nwerc.eu>

World Finals

- gibt es seit 1977
- derzeit 120 Teams
- die vorderen Plätze werden inzwischen von China und Russland dominiert
- „Dabei sein ist alles“;)



ACM-ICPC
World Finals
June 22 - June 26
2014
Ekaterinburg
host Ural Federal University

IBM event sponsor

vgl <http://icpc.baylor.edu/>

Was für Aufgaben werden gestellt?

- Knifflige Probleme, die aber meist recht kurz lösbar sind
- Wahl des passenden Algorithmus mit der passenden Komplexität sehr wichtig
- Schema der Aufgaben:
 - Kurze einleitende Geschichte
 - Beschreibung des Problems
 - Definition der Eingabedaten
 - Definition der Ausgabedaten
 - Kleine Beispiele
- Eingabe und Ausgabe über Standardstream (stdin/stdout - stderr wird ignoriert)

IKEA

Building an IKEA kitchen is not as trivial as you might think. The construction manual is incomprehensibly written and some screws are always missing. However, the biggest problem is that the packages are always in the wrong place. The carrier placed all packages as one big stack in the corner of your kitchen. Unfortunately, you want to start the construction of your new kitchen in exactly that corner. Thus, you need to move the stack away.

However, the packages are very heavy. In one step, you can only move the topmost package from one stack to another stack. Furthermore, the carrier sorted the packages from heavy (bottom) to lightweight (top). You also do not want to place a heavier on a lighter package. There is enough space in the kitchen to create two more stacks.

Input

The first line denotes the number of test cases $1 \leq t \leq 100$. Each of the following t lines contains one integer p ($1 \leq p \leq 50$), the number of packages in the stack.

Output

For each test case, print one line containing the number of moves required to move every package from the original stack in the corner to a new stack (probably not in that corner) using at most one “helper” stack.

Sample Input

```
2
3
42
```

Sample Output

```
7
4398046511103
```

Wie werden die Aufgaben ausgewertet?

- Abgabe der Lösung (als Quelltext) erfolgt über ein Webinterface (oder manchmal auch zusätzlich via Command-Line)
- Rückfragen (z.B. unklare Problemstellungen) ebenfalls via Web - diese sind für alle einsehbar
- Auswertung erfolgt automatisch anhand unveröffentlicher Testcases
- magere Rückmeldung, z.B.
 - **CORRECT** - Korrekt gelöst
 - **WRONG-ANSWER** - Fehler bei einem Testcase
 - **COMPILER-ERROR** - Fehler beim Übersetzen
 - **TIMELIMIT** - Ausführung braucht zu lange
 - **RUN-ERROR** - z.B. segfault bei Ausführung
- eine Jury prüft die Abgaben aber sicherheitshalber nochmals nach

[overview](#) [scoreboard](#)

Sun 26 Jun 2011 13:29:23 CEST

#	AFFIL.	TEAM	SCORE	A	B	C	D	E	F	G	H	I	J	
?		testteam	8	486	1 (51 + 0)	0	1 (52 + 0)	3 (54 + 40)	1 (55 + 0)	1 (56 + 0)	3	1 (58 + 0)	1 (59 + 0)	1 (61 + 0)

Submissions

Select file...

problem

language

submit

cancel

Add another file

time problem lang result

01:05	A	HASKELL	TOO-LATE
21:14	D	CPP	RUN-ERROR
21:04	A	C	RUN-ERROR
21:47	J	CPP	CORRECT
21:47	J	CPP	CORRECT
21:47	J	CPP	RUN-ERROR
21:47	J	CPP	WRONG-ANSWER
21:47	J	C	CORRECT
21:47	J	CPP	CORRECT
21:46	J	CPP	CORRECT
21:46	J	JAVA	CORRECT
21:46	J	CPP	CORRECT
21:46	I	CPP	WRONG-ANSWER
21:46	I	CPP	CORRECT
21:45	I	CPP	CORRECT
21:45	I	C	WRONG-ANSWER
21:45	I	CPP	WRONG-ANSWER

Clarifications

time from to subject text

12:02	Jury	All	problem F	The weakest link is the region with the minimum number of armies *after* one tu...
11:30	Jury	All	problem H	Prices are integers (and so is the number of shares).
11:18	Jury	All	general	We have decided to extend the contest by 15 minutes. Therefore the end time is ...
10:52	Jury	All	general	If you want to set your keyboard layout to German, then you can do this by execu...

Clarification Requests

No clarification requests.

request clarification

Bildquelle: https://www.progsoc.org/wiki/UTS_Programming_Competition_2013

Und wie werden die Teams gewertet?

- Wichtigste Kategorie ist die Anzahl der korrekt gelösten Aufgaben
- Als weiteres Kriterium bei Gleichstand wird die Abgabezeit der korrekten Lösungen herangezogen
- Pro falscher Abgabe gibt es 20 Strafminuten (allerdings nur, wenn diese irgendwann auch korrekt gelöst wurde)
- Das Scoreboard ist für jeden einsehbar - oft lassen sich dadurch einfache Aufgaben erkennen ;)
- Aber 1h vor Ende wird das Scoreboard eingefroren - bis zur Siegerehrung

Scoreboard NWERC 2011

final standings

#	AFFIL.	TEAM	SCORE	A ●	B ●	C ●	D ●	E ○	F ●	G ●	H ●	I ●	J ●	
1		geen.opdracht5	7	898	1 (22 + 0)	1 (69 + 0)	2 (97 + 20)	4 (277 + 60)	1 (44 + 0)	0	1 (134 + 0)	1 (175 + 0)	0	0
2		Wave of Technology	7	1153	3 (71 + 40)	1 (103 + 0)	1 (127 + 0)	2	1 (26 + 0)	0	9 (292 + 160)	2 (115 + 20)	1 (199 + 0)	0
3		Joy	7	1217	2 (106 + 20)	1 (148 + 0)	2 (198 + 20)	0	1 (38 + 0)	0	3	2 (240 + 20)	1 (88 + 0)	4 (279 + 60)
4		The team name is too long to .	6	990	2 (46 + 20)	1 (131 + 0)	1 (139 + 0)	0	1 (103 + 0)	0	3 (205 + 40)	3 (266 + 40)	0	0
5		Team Squirrel	6	990	4 (138 + 60)	1 (48 + 0)	1 (34 + 0)	2 (260 + 20)	1 (88 + 0)	0	0	4 (282 + 60)	0	0
6		segFAULT_	6	1026	3 (179 + 40)	2 (157 + 20)	1 (80 + 0)	1	1 (42 + 0)	0	0	2 (272 + 20)	1 (216 + 0)	0
7		Lambdabamserne	5	539	1 (35 + 0)	1 (68 + 0)	2 (164 + 20)	0	1 (92 + 0)	0	0	1 (160 + 0)	3	0
8		Jacobs One	5	698	1 (84 + 0)	1 (172 + 0)	1 (21 + 0)	0	1 (140 + 0)	0	0	3 (241 + 40)	0	0
9		Cambridge 1	5	728	3 (109 + 40)	1 (100 + 0)	1 (40 + 0)	0	1 (69 + 0)	0	0	3	5 (290 + 80)	0
10		First Ship	5	786	3 (287 + 40)	1 (163 + 0)	1 (83 + 0)	0	1 (56 + 0)	0	2	2 (137 + 20)	0	0
11		Geen Commentaar	5	827	3 (271 + 40)	1 (79 + 0)	1 (54 + 0)	4 (298 + 60)	1 (25 + 0)	0	0	1	0	0
12		Edinburgh	5	922	4 (287 + 60)	1 (160 + 0)	1 (241 + 0)	0	1 (35 + 0)	0	0	1 (139 + 0)	0	0
13		false'); DROP TABLE teams; --	5	966	2 (261 + 20)	1 (182 + 0)	1 (256 + 0)	0	1 (51 + 0)	0	0	1 (196 + 0)	0	0
14		Karlsruhe International Team	5	1066	3 (146 + 40)	3 (256 + 40)	3 (229 + 40)	1	1 (25 + 0)	0	2	1 (290 + 0)	1	1
15		Syntax Error	4	556	3	2 (203 + 20)	1 (44 + 0)	0	1 (67 + 0)	0	0	1 (222 + 0)	0	0

Auswahl der Aufgaben

- am Anfang: Querlesen der Probleme, um leichte Aufgaben zu finden
- Rangliste gibt meist sehr guten Anhaltspunkt für Schwierigkeiten der Aufgaben:
 - wie oft wurde die Aufgabe gelöst?
 - wurde die Aufgabe bereits früh gelöst?
 - viele Fehlsubmissions vor korrekt: Grenzfälle nochmal durchgehen
- individuelle Stärken berücksichtigen
- Unstimmigkeiten sollten **vor** Bearbeitung einer Aufgabe geklärt werden
- bei Unsicherheiten kann Paarprogrammierung sinnvoll sein
- Programme bereits auf Papier coden
- nicht zu viele Aufgaben anfangen

Was tun im Fehlerfall?

- so wenig Rechnerzeit verschwenden wie möglich
- bei SEGFAULT Debugger (gdb / valgrind) verwenden
- zusätzliche Testcases generieren
- Randfälle prüfen
- auf Überläufe oder Rundungsprobleme prüfen
- Laufzeit prüfen bzw. Endlosschleifen lokalisieren
- Aufgabe im Team besprechen
- nach erstem Fehler nicht sofort resubmiten

Idee

Türme von Hanoi-Problem mit p Scheiben



Idee

Türme von Hanoi-Problem mit p Scheiben

Weitere Überlegungen

- Durchsimulieren dauert (viel) zu lange
- Formel ist (*grübel*) $2^p - 1$
- Grenzfall: p ist maximal 50
 - ⇒ Ausgabe maximal ca. 2^{50}
 - ⇒ passt nicht in 32 bit Integer (2^{32})
 - ⇒ 64 bit Datentyp nehmen (long long in C/C++, long in Java)
- bitwise shift geht schnell und berechnet haargenau was wir wollen: $1 \ll p \Leftrightarrow 2^p$

in C

```
1  #include <stdio.h>
2
3  int main(){
4      int i,t;
5      scanf("%d\n", &t);
6      for (i=0; i<t; ++i) {
7          int p;
8          scanf("%d\n", &p);
9          printf("%lld\n", ((1LL << p) - 1));
10     }
11 }
```

in C++

```
1  #include <iostream>
2
3  using namespace std;
4
5  int main() {
6      int t;
7      cin >> t;
8      for (int i=0; i<t; ++i) {
9          int p;
10         cin >> p;
11         cout << ((1LL << p) - 1) << endl;
12     }
13 }
```

in Java

```
1 import java.util.Scanner;
2
3 public class IKEA {
4     public static void main(String[] args) {
5         Scanner in = new Scanner(System.in);
6         int t = in.nextInt();
7         for (int i = 0; i < t; ++i) {
8             int p = in.nextInt();
9             System.out.println((11 << p) - 1);
10        }
11    }
12 }
```

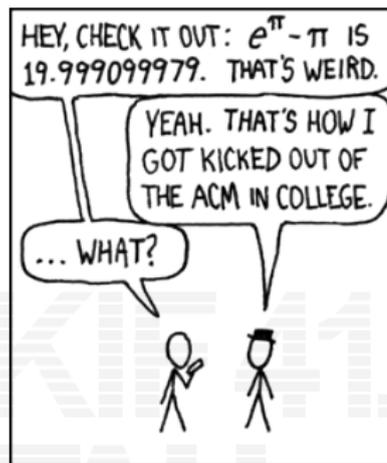
Interesse?

- Für die Dauer der KIF41.5 läuft eine Trainingsumgebung
 - Zugangsdaten gibt's jederzeit von rupran oder thelazt
 - Login auf <http://icpc.cs.fau.de/domjudge/team/>
 - Scoreboard auf <http://icpc.cs.fau.de/domjudge/>
- An der eigenen Hochschule: Gruppe gründen und
 - im Januar '14 am Erlanger Wintercontest teilnehmen
 - im Juni '14 (oder so) am German Programming Contest (kurz GCPC) teilnehmen
 - im November '14 am NWERC teilnehmen?
 - im Frühjahr '15 die WorldFinals rocken ;)

weitere Details für Wintercontest und GCPC auf <http://icpc.cs.fau.de/> - am besten mal Daniel Brinkers

(icpc@i2.cs.fau.de) / Lehrstuhl für Informatik 2 (Programmiersysteme) der FAU lieb fragen ;)

Noch Fragen?



DURING A COMPETITION, I TOLD THE PROGRAMMERS ON OUR TEAM THAT $e^\pi - \pi$ WAS A STANDARD TEST OF FLOATING-POINT HANDLERS -- IT WOULD COME OUT TO 20 UNLESS THEY HAD ROUNDING ERRORS.



von <http://xkcd.com/217/>

