

# 대학생 프로그래밍 대회의 동향과 변화

한국외국어대학교 | 김희철\*

한국과학기술원 | 이현섭\*\*

## 1. 서론

국제대학생 프로그래밍대회인 ACM-ICPC[1] 한국대회는 2000년 처음 KAIST에서 아시아 지역대회로서 개최되었고, 2001년부터는 정보통신부에서 주최하는 대학생 프로그래밍 경시대회와 겸하여 개최되고 있다. 이 대회는 대학생들의 창의적인 문제 해결능력과 이를 구현하는 프로그래밍 능력을 경시하는 팀 경시대회로서 하나의 팀은 3명으로 구성되며 한 대의 컴퓨터를 사용하여 주어진 시간에 출제된 8-10개의 문제들에 대한 해를 구하는 프로그래밍 능력을 경시한다. 학생들은 각 문제에 대하여 주어진 수행시간이내에 올바른 답을 구하는 프로그램을 만들어야 한다. 따라서 경시대에서는 문제에 대한 정확한 답을 구해야 함과 동시에 효율적인 프로그램을 요구한다.

현재 아시아에서는 한국을 비롯하여 중국, 일본 등 13개 지역에서 국제대학생 프로그래밍 지역대회를 개최하고 있다. 한국은 2000년 국제대학생 프로그래밍 지역대회를 개최한 이후, 2006년까지 매년 대회가 개최되면서 많은 변화와 발전을 해 오면서 우리나라의 대학교 프로그래밍 교육에 많은 기여를 해왔다.

본고에서는 지난 7년간 국제대학생 프로그래밍 한국대회의 동향과 변화를 살펴본다.

## 2. 역대 대회의 특성과 변화

2000년 40개 대학, 55팀으로 시작한 국제대학생 프로그래밍 한국대회는 2006년 62개 대학, 207팀이 참가하는 등 규모면에서의 성장만이 아니라, 참가 학생들의 프로그래밍 능력의 향상과 함께 대학생 프로그래밍 경시대회로는 가장 크고 권위 있는 대회로 자리잡았다.

초창기의 대회에서는 상위 몇 개 팀을 제외하고는

참가 팀들이 해결한 문제들이 많지 않았으나, 해를 거듭하면서 많은 학생들이 관심을 가지면서 참가팀 수도 증가하고 문제해결 능력 수준도 높아지고 있다. 경시에 출제되는 문제들도 직관적인 해결 방법을 찾을 수 있는 것 보다는 응용한 형태로서 해결방법 뿐만 아니라 문제의 특성을 잘 이해하는 것을 요구하는 형태로 발전되어 왔다.

2002년 대회부터는 참가팀수가 늘어남에 따라 예선을 거쳐 본선대회의 참가팀을 선발하였다. 2004년 대회부터는 장소를 서울로 옮겨 개최되어 왔다.

### 2.1 경시환경 및 프로그램 채점방법

경시대회에 참가한 각 팀에는 PC 한 대씩이 주어지며, 사용할 수 있는 프로그래밍 언어는 C/C++, Java로, 모든 팀이 동일한 환경에서 경시에 임할 수 있도록 준비가 된다. 이들 PC들은 한곳에 설치되고, 또한 채점 서버에 연결이 되어있다. 출제된 문제들에 대하여 팀의 구성원들이 함께 협동하여 해결할 수 있다. 팀이 문제의 해를 구하는 프로그램을 완성하여 서버에 제출하면 서버에서 이 프로그램을 채점하여 그 결과를 다시 보내준다. 제출 프로그램에 대한 채점은 출제문제에 대하여 서버에 미리 준비한 모든 테스트 경우에 대하여 제한시간이내에 올바른 답을 출력해야 해결한 것으로 인정받는다. 문제에 따라서는 효율적인 방법을 요구하기 위해서 주어진 환경에서 효율적인 메모리 관리를 요구하기도 한다. 제출 프로그램이 틀린 답을 출력하든지 혹은 제한시간 등 요구조건을 만족하지 않을 경우 오답으로 처리한다.

각 문제에 대하여, 팀이 해결한 완전한 프로그램을 제출했을 때에는 이 문제 해결에 대한 별점이 다음과 같이 주어진다: 별점 = {대회 시작 후 이 문제를 해결하는 완전한 프로그램을 제출하는데 걸린 시간(분) + (그 문제에 대한 오답 프로그램 제출 횟수) × 20}. 예를 들어서 A번 문제를 풀어서 제출했다가 2번 틀리고, 경시시작이후 30분이 되어서 맞추게 되면 (30 + 2 ×

\* 정회원

\*\* 학생회원

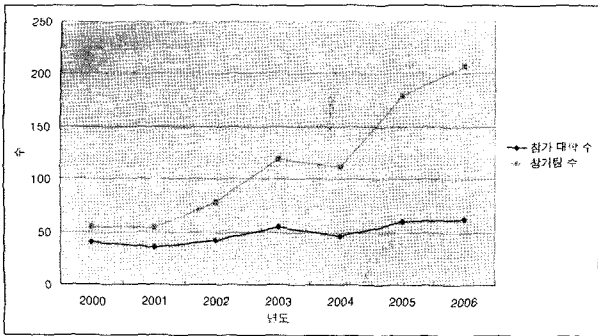


그림 1 대회별 참가팀 수

20) = 70점의 별점이 주어진다. 따라서 문제를 빨리 풀수록, 틀리지 않고 맞출수록 낮은 별점을 얻게 된다.

경시에서 팀 순위는 해결한 문제의 수에 따라 정해진다. 해결한 문제 수가 많은 팀이 순위가 높고, 해결한 문제 수가 같을 경우 전체 별점이 작은 팀이 순위가 높다.

## 2.2 참가팀 수

2000년도에 40개 대학에 55개 팀(외국 3팀 포함)이 참가하였다. 그 이후 참가팀 수가 꾸준히 증가하여 2005년에는 60개 대학에서 179팀이, 2006년에는 62개 대학 207개 팀(외국 6팀 포함)이 참가하였다. 그림 1에서 보는바와 같이 참가팀의 수는 4배 가까이 증가하였고 참가 학교의 수는 50% 증가하여, 전체 대학교의 1/3 정도의 학교에서 참가하고 있다. 그림 1에서 볼 수 있듯이, 참가하는 학교는 매년 꾸준히 증가하고 있는 추세이다.

## 2.3 인터넷 예선대회

국제대학생 프로그래밍 한국대회에 참가하는 팀 수는 대회 장소의 크기, 채점방법 등에 의하여 60여개로 정해져 있다. 2002년 대회부터는 참가팀이 많아짐에 따라 인터넷으로 예선 대회를 개최하여 본선 대회에 출전하는 팀을 선발하였다. 인터넷 예선대회는 참가하는 대학에서 경시환경을 만든다. 참가하는 모든 팀들은 같은 시각에 경시를 시작하고, 공정성을 위하여 각 참가 팀의 코치 감독 하에 경시가 진행된다. 문제는 온라인으로 출제하여, 참가 팀들은 문제의 해를 구하는 자신의 프로그램을 채점 서버에 제출하면 채점서버에서 프로그램의 수행시켜 그 결과를 알려준다. 인터넷 예선대회를 치름으로써 본선에 올라온 학생들은 일정 수준 이상의 문제를 해결할 수 있는 능력을 갖고 있었고 채점 시스템도 잘 이해하여 대회의 수준은 한층 높아졌다.

컴퓨터 전공 대학생들에게 문제해결방법과 프로그

래밍의 중요성을 인식시키고 또한 학생들의 관심을 높이기 위한 방법으로 교내경시대회를 개최하는 대학교들이 증가하고 있다. 교내 대회의 문제 출제와 프로그램에 대한 채점 등의 작업을 돕기 위해서 대학생 프로그래밍 대회의 인터넷 예선 대회를 이용하여 교내 경시를 할 수 있도록 지원하고 있다. 가톨릭대, 국민대, 경북대, 숭실대 등이 인터넷 예선 대회를 활용하여 교내 대회를 개최하고 있다.

국내 대학의 팀들만 참가하는 초창기의 인터넷 예선 대회에서는 문제를 한글로만 출제하였다. 인터넷 예선 대회를 통해 선발된 팀은 외국에서 참가하는 5팀 내외의 외국 대학팀과 함께 국제대학생 프로그래밍 한국대회에 참가하게 되는데, 이때 모든 본선 문제들은 영어로 기술된다. 예선을 통과한 본선 진출 팀들이 본선 대회에서 출제되는 영어 문제에 대한 적응력을 높이도록 하기위하여 2005회 대회부터는 인터넷 예선 대회에서부터 영어로 기술된 문제도 함께 출제하고 있다. 이는 대학생들의 영어에 대한 준비를 하도록 함으로서 공학도가 가져야 할 어학능력을 높이는 데 도움이 되고 있다.

## 2.4 출제문제 분석

### 2.4.1 출제경향

국제대학생 프로그래밍 한국 역대대회에 출제된 문제들을 분석해보면 기본적인 자료구조에 대한 문제에서부터 그래프 이론을 응용한 문제, 기하학, 조합 등 다양한 분야에서 문제들이 출제되어 왔다.

대회에서 출제되는 문제들에 대한 효율적인 프로그램을 만들기 위해서는 자료구조/알고리즘 교과목에서 가르치는 기본적인 자료구조를 이해하고 이를 구현할 수 있어야 하고, 또한 문제 해결 방법으로서 동적 프로그래밍(dynamic programming), 욕심쟁이 알고리즘(greedy algorithm), 분할정복 알고리즘(divide & conquer algorithm), 탐색방법 등의 알고리즘 설계기법과 계산기하학과 관련한 기본적인 알고리즘을 이해하고 응용할 수 있어야 한다. 초창기 대회와 비교하여 참가하는 학생들의 실력이 점진적으로 향상됨에 따라 출제되는 문제들은 전형적이고 기초적인 형태의 문제들의 비율은 낮아지고, 문제를 이해하고 이를 분석한 후 해결 방법을 찾아가는 능력을 요구하는 문제들의 비중이 높아지고 있다. 지금까지 출제된 문제들을 유형별로 분석하면 다음 표와 같으며 역대 대회의 문제들은 대회 홈페이지 <http://acm.kaist.ac.kr>에서 확인할 수 있다.

표 1 역대 대회 출제문제의 출제 경향

	2001	2002	2003	2004	2005	2006
동적 프로그래밍	2	1	2	2	1	1
그리디 알고리즘	1	0	0	0	0	0
기본적 자료구조	0	2	1	1	1	0
탐색 기법	1	0	0	0	2	0
그래프 이론	1	2	2	3	2	3
계산기하학	1	0	2	1	1	3
Enumeration	0	0	1	1	0	1
수학적 특성 파악	2	3	1	1	3	2
계	8	8	9	9	10	10

### 2.4.2 출제문제들의 난이도

출제되는 문제는 초창기에는 8문제를 출제하였다. 요즘은 10문제가 출제되고 있다. 이들 중 쉬운 문제들을 1~2개 출제하고, 중간 난이도의 문제를 4~5개, 고 난이도의 문제를 2~3개를 출제한다.

해가 거듭해갈수록 대회에 관심을 갖고 준비하는 학생들이 많아짐에 따라 학생들의 수준은 눈에 띄게 높아져 가고 있다. 이에 따라 난이도가 높은 문제들의 비중이 높아져 가고 있으며 다양한 유형의 문제를 출제하기 위해서 문제 수가 10문제까지 늘어났다. 제한된 시간으로 인해 많은 문제수가 학생들에게 부담이 되기도 하겠지만 짧은 시간에 문제를 분석하고 문제의 난이도를 판별하는 능력까지도 갖추어야 좋은 성적을 얻을 수 있다.

출제되는 문제들은 초창기에는 난이도가 쉬운 문제부터 어려운 문제들로 나열하였다. 따라서 참가팀들이 문제들의 나열 순서에 의하여 난이도를 짐작할 수 있었고, 해결하는 전략을 세울 수 있었다. 그 이후 2004년도부터는 아주 쉬운 문제는 처음에 오도록 하고, 중간 난이도 이상의 문제들은 난이도 순서 대신 임의의 순서로 나열하였다. 이는 문제들을 읽고 난이도를 판별하는 능력을 가지도록 요구하고 있다.

### 2.4.3 대회별 출제문제들의 해결을

초창기 대회에서는 정확하면서도 수행시간이 적게 걸리는 프로그램을 작성해야 하는 것과 실시간으로 채점하는 시스템은 상위 몇몇 팀을 제외한 많은 학생들에게는 어렵게 느껴졌지만 그 이후로 전체적인 문제 해결 수준이 높아져 가고 있다.

그림 2에서는 그동안 출제되었던 문제들의 난이도를 상, 중, 하로 나누어서 지난 대회의 해결율(해결한

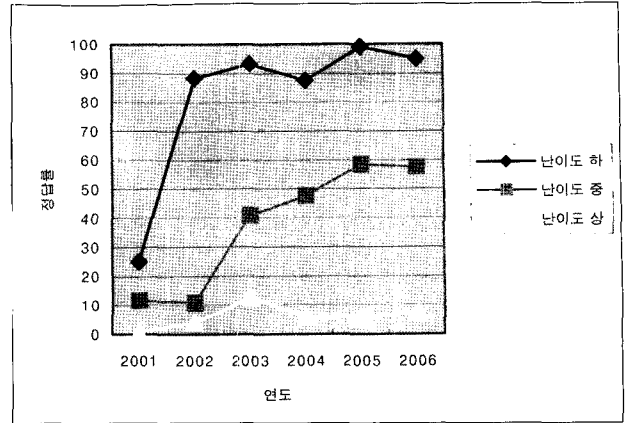


그림 2 역대 본선 대회 난이도별 해결율

팀 수 / 전체 팀 수)의 변화를 볼 수 있다. 각 대회별로 난이도의 차이가 있고, 또 출제된 문제들 유형이 다르지만 2002년을 기점으로 쉬운 문제들은 대부분의 참가팀이 풀게 된 것을 볼 수 있다. 2001년에는 인터넷 예선이 없이 참가 신청을 한 팀은 모두 대학생 프로그래밍 경시대회에 참가할 수 있었기 때문에 대회 진행 방식이 익숙하지 않은 팀들은 쉬운 문제조차도 해결하는데 어려움이 있었다. 하지만 2002년부터 인터넷 예선이 진행됨에 따라 대회 진행 방법에 한결 익숙해지고, 어느 정도 수준이상인 팀들이 본선에 진출함에 따라 쉬운 문제는 대부분 풀 수 있게 되었다.

매년 출제되는 문제들의 난이도는 일정 수준 유지하고 있는데, 중간 난이도 문제의 정답률이 2003년부터 증가하여 50~60% 정도의 팀들이 중간 난이도 문제를 해결하였다. 이는 대회를 준비하고 참가하는 팀들의 수준이 높아졌음을 보여주는 결과이다. 반면 난이도가 높은 문제의 경우 참가팀들의 정답률이 높지 않은데, 난이도가 높은 문제에 대한 해결 능력을 높이기 위해서는 팀별 자체적인 연습과 함께 문제해결 방법을 구하는 체계적인 교육이 필요하다.

### 2.4.4 2006년도 출제문제 분석

그동안 출제된 문제들은 [2]에서 볼 수 있다. 여기서는 2006년도에 출제된 문제들과 그 해결방법[3]을 소개하고, 또한 이들 문제들에 대하여 참가팀들의 해결 정도를 분석한다. 2006년도에는 문제 A부터 문제 J까지 총 10문제가 출제되었다.

#### 문제 A, Digital Clock

디지털시계의 시각은 하나의 자연수로 표시될 수 있다. 두 시각이 주어 질 때, 그 사이에 3의 배수가 몇 개나 있는지를 계산하는 문제이다.

해결방법: 루프를 통하여 해결할 수 있는 간단한 문제이다.

문제 B. DNA Consensus String

길이가 같은 DNA 염기서열들이 주어질 때에, 주어진 염기서열들을 대표할 수 있는 새로운 DNA 염기서열을 구하는 문제이다.

해결방법: 이 문제에서는 주어진 염기 서열들 간의 Hamming 거리의 합이 가장 작은 염기서열을 구하도록 하고 있다. 매 i번째 염기들 중 가장 많이 나오는 것을 선택하여 구할 수 있다.

문제 C. Grid Panel

그리드 패널(panel)에 구멍이 뚫려 있을 때, 이 구멍들을 포함하는 직교 다각형 중 문제에서 요구하는 조건을 만족하는 가장 작은 것의 넓이를 계산하는 문제이다.

해결방법: 보통의 평면(plane) sweep 알고리즘으로 해결이 가능하다.

문제 D. Travel

주어진 그래프에서 꼭 방문하고 싶은 두 간선을 지나는 단순 사이클이 존재하는지를 묻는 문제이다.

해결방법: 이중연결(biconnected) 요소들을 구하여 방문하고 싶은 두 간선이 하나의 이중연결 요소에 포함되는지 검사하면 된다.

문제 E. Roommate

같은 기숙사에 사는 두 룸메이트가 공유하는 여러 가지 물품에 대해 서로간의 사용시간을 스케줄링 하는 문제이다.

해결방법: 특수한 케이스의 그래프에서의 최단경로 문제로 변환하여 해결 가능하다. 혹은, 동적프로그래밍으로도 가능하다.

문제 F. Gas

주어진 약품 C의 양을 잘 조절하여 각 시험관에서 같은 양의 기체 G가 나오도록 해야 한다. C의 전체 양 M이 주어지며, 각 시험관에서 생산되는 기체 G의 양은 선형 함수로 주어진다. 이 문제에서는 특별히 유리수 연산을 필요로 하고 있다.

해결방법: 하나의 시험관에서 생산되는 기체의 양을 계산하여 각 시험관에 주어지는 선형함수를 풀면 된다.

문제 G. Tree

평면상에 점들이 주어져 있고, 각 점에는 분지수(degree) 값들이 매겨져 있다. 이 문제는 각 점의 분지수 값이 트리를 이룰 수 있는 조건을 만족할 때에 그 트리를 실현하는 문제이다.

해결방법: Convex hull을 이용하여 해결할 수 있다.

문제 H. Period

두 개의 문자열이 주어졌을 때에, 하나가 다른 하나의 k-근사 주기(k-approximate period)가 되는 k 중 최소 값을 찾는 문제이다.

해결방법: 이 문제를 풀기 위해서는 먼저 두 문자열 사이의 편집거리(edit distance)를 계산할 줄 알아야 한다. 이것은 평범한 동적 프로그래밍으로 가능하다. 그 이후에 고정된 정수 k가 주어졌을 때에 입력으로 주어진 두 개의 문자열이 k-근사 주기 관계를 만족하는지 판별하는 알고리즘을 이용하여 전체 문제를 해결할 수 있다.

문제 I. Fire Tower

1차원 지형이 다각형 체인(polygonal chain)으로 주어질 때에, 그 지형의 모든 점을 볼 수 있는 가장 길이가 짧은 탑을 지으려고 한다. 그 높이는 얼마일까? 다각형 체인의 각 선분으로부터 그 선분이 보이는 점들의 집합은 그 선분이 정의하는 반평면(half-plane)으로 나타난다. 이것들의 교집합에 탑의 꼭대기가 위치해야 하는 것이다. 문제에서는 탑의 길이를 최소화 하는 것이기 때문에 지형을 염두에 두어야 한다.

해결방법: 주어진 다각형 체인의 연장선으로 계산할 수 있는 볼 수 있는 반평면(half-plane)의 교집합 영역의 가장 작은 y 좌표를 찾으면 된다.

문제 J. Log Jumping

이 문제는 단위-구간 그래프(unit-interval graph)에서 해밀톤 사이클 (Hamiltonian cycle)을 갖는 최대 induced 그래프를 찾는 문제이다.

해결방법: 단위-구간 그래프에서 해밀톤 사이클이 있는지를 판별하는 것은 이중연결 요소 알고리즘을 통해 쉽게 가능하며 같은 방법으로 이 문제도 해결할 수 있다.

위의 문제들에 대하여 본선에 참가한 61개 팀들이 해결한 문제들을 분석하면 표 2와 같다.

표 2 2006년도 대회 문제의 해결율

문제	해결한 팀 수	해결율 (%) (해결한 팀 수)/(전체팀수)
A	60	98.36
B	61	100
C	5	8.16
D	10	16.39
E	0	0.00
F	36	59.01
G	0	0.00
H	6	9.83
I	6	9.83
J	7	11.47

출전한 팀들은 대부분 쉬운 문제에 대한 해는 구하였다. 앞에서 언급한 바와 같이, 문제들의 순서가 난이도에 의한 것이 아니기 때문에, 어떤 문제를 먼저 시도하느냐에 따라 팀 성적에 있어서 많이 차이가 날 수 있다. 어려운 문제를 먼저 시도하면 이 문제를 해결하는데 많은 시간을 소비함으로써 결과적으로 제한 시간에 해결하는 문제들의 개수에 영향을 미치게 된다. 따라서 팀이 좋은 성적을 얻기 위해서는, 출제되는 문제에 대한 분석을 통하여 문제의 난이도를 판단할 수 있어야 하고, 해결하는 문제들의 순서도 고려해야 한다.

## 2.5 기타

2001년 대회 때부터 국제대학생 프로그래밍 한국대회는 정보통신부 주최 대학생 프로그래밍 대회와 함께 진행되어 왔다. 대회규모가 커지고 인터넷 예선대회까지 개최하게 됨에 따라, 2004년 대회부터는 산업체의 후원을 받아 인터넷 예선 대회와 본선 대회의 진행을 개선할 수 있었다. 인터넷 예선 대회도 성적에 따라 지역별 우수 팀과 우수 학교를 시상할 수 있게 되어 참가팀과 참가 학교의 참가 의욕을 북돋아줄 수 있었다.

국제대학생 프로그래밍 한국대회에서 우수한 팀은 ACM-ICPC 세계 본선 대회에 참가하게 된다. ACM-ICPC 세계대회는 각 지역 대회의 우승팀들이 참가하여 경시를 하는데, ACM-ICPC 한국대회가 개최되기 전에는 우리나라 팀은 아시아의 다른 지역대회에 참가하여 우승함으로써 세계대회 본선에 참가하였다. 우리나라 팀은 1996년에 처음 세계대회에 참가하여 1997년에 10위에 입상하였고, 2001년에는 8위에 입상하였다. 이후 매년 세계대회에 참가하는 우리나라 팀들은 10~20위 사이의 성적을 거두고 있다. 대회를 통한 동기 부여와 참가 학생들의 높은 관심으로 대회에서 요구하는 창의적 문제 해결 능력, 협동적 프로그래밍 능력 등이 어느 수준까지 성장하였다. 우리나라 팀이 세계 대회에서 최고 수준에 도달하기 위해서는 학생들을 위한 체계적인 이론교육과 함께 팀을 구성하는 학생들이 다양한 문제에 대하여 그 해결방법을 함께 토론하여 효율적인 방법을 찾고 이를 검증하는 연구 자세가 필요하다.

## 3. 결론

우리나라 미래 IT를 이끌어갈 IT 전문 인력으로 성장하기 위해서는 창의적 문제 해결 능력, 협동적 프로그래밍 능력을 가져야 한다. 국제대학생 프로그래밍 대회는 참가학생들이 이러한 능력을 갖추도록 하는데 지속적으로 역할을 해왔다. 또한, 참가학생들은 대회에서 다른 대학의 학생들과 경쟁함으로써 자신의 잠재적 능력을 개발하고 발전시키는데 자극을 받는 등의 교육적 효과도 거두었다. 미래 IT 리더를 꿈꾸는 학생들의 최고를 향한 성장을 위해서는 대회를 통해 동기를 얻은 학생들이 더욱 성장하고 발전할 수 있도록 창의적이고 협동적인 문제해결능력 개발을 위한 체계적인 교육에 대한 뒷받침이 필요하다.

## 참고문헌

- [1] <http://icpc.baylor.edu>.
- [2] <http://acm.kaist.ac.kr>.
- [3] 제6회 대학생프로그래밍 경시대회 보고서, 한국정보과학회, 2006년



### 김희철

1980 서울대학교 계산통계학과(학사)  
 1982 한국과학기술원 전산학과(석사)  
 1987 한국과학기술원 전산학과(박사)  
 1987~현재 한국외국어대학교 컴퓨터및정보통신공학부 교수  
 1997~1998 미국 Michigan State University 방문교수

관심분야: 알고리즘, 상호연결망, 그래프이론 등  
 E-mail : hckim@hufs.ac.kr



### 이현섭

2001 한국과학기술원 전산학과(학사)  
 2003 한국과학기술원 전산학과(석사)  
 2003~현재 한국과학기술원 전산학과 박사과정  
 관심분야: 알고리즘, 상호연결망, 그래프이론 등  
 E-mail : haru@jupiter.kaist.ac.kr