2

ACM 국제 대학생 프로그래밍 대회(ICPC)와 그 교육적 효과

부산대학교 ▌조 환 규*

1. 경시대회와 프로그래밍 교육

모든 컴퓨터 관련 교육기관에서는 좋은 프로그래머를 배출하기 위하여 무던히 애를 쓴다. 또한 산업계에서는 좋은 프로그래머를 찾기 위하여 항상 노력하고 있다. 그런데 어떤 프로그래머가 좋은 프로그래머인가를 따지고 생각해보면 문제는 단순하지 않다. 이는 마치 가장 훌륭한 야구선수는 어떤 사람인가를 판단하는 문제와도 유사하다. 투수를 제외한 야구선수들은 매우 많은 지표를 가진다. 타율, 타점, 출루율, 장타율, 홈런, 실책 등. 따라서 상황 상황에 따라 야구선수를 선별하는 기준을 조금 달라질 수밖에 없듯이 프로그래머도 각 상황에 따른 개별의 기준으로 평가할 수밖에 없을 것이고 여기에 경시형 프로그래밍 대회는 주요한 역할을 한다. ICPC 프로그래밍 대회는 특정한 기준으로 참가자들을 선형화하여 몇 참가자들에게 상을 주는 방식으로 진행된다. 이른바 경시대회라는 것이 이 형식인데, 여기에는 여러 비판도 있다.

예를 들어 이전 한국 정보올림피아드에 운영위원으로 할 때의 일이다. 한 학생, 아마도 중학생으로 기억하는데, 그 학생의 답이 완전히 틀려서 0점을 받았는데 그 학생의 학부모와 지도교사(대회 참가를 위하여 교육시킨 학원장으로 추측되는) 되시는 분이 와서 강력하게 항의를 했다. 항의의 요지는 두 가지였다. 자기 아들은 분명히 프로그램을 맞게 구성하여 틀릴 수가 없다는 것이고 다른 하나는, 설사 답이 약간(?) 틀렸다고 해도 그렇게 0점을 주면 안 된다는 주장을 했다. 설명하시길 교육이란 과정이 중요한데 학생이 프로그래밍을 한 과정을 자세히 살펴보고 그 과정이 적절하면 점수를 충분히 줘야 한다고 요구했다. 이런 아수라장 속에 다행히도 당시 운영위원장을 맡으셨든 교수님의 논리적인 설명으로 사태는 수습되었고 모든 학생의 성적은 오류 없이 확정된 후 공지되었다.

필자는 이런 현상을 〈경진대회의 비극〉이라고 부르고 싶다. 물론 중학생이 정보올림피아드에서 입상하는 것이 과학고 입시에서 절대적인 지름길을 확보하는 방편이 빚어낸 해프닝으로도 볼 수 있지만 여러 채점위원 교수님들은 불편하기 짝이 없었다.

경진대회는 컴퓨터 과학뿐만이 아니라 모든 과학, 예술분야 어디에나 있다. 특히 음악계에서 경진대회는 매우 중요하다. 연습 때 아무리 잘 쳐도 본선에서 조그만 실수를 한다면 무대 뒤에서 탄식의 눈물을 흘려야 한다. 하지만 Schiff와 같이 정규 콩쿠르를 통하지 않고서도 얼마든지 국제적 수준의 피아니스트가 될 수도 있는 반면 콩쿠르 우승 뒤에 바로 명멸해버린 음악가도 얼마든지 볼 수 있다. 모든 경진대회는 이런 경직성을 피할 수 없다. 하지만 ACM-ICPC와 같은 경시대회가 프로그래머의 능력을 측정하기에 적절치 못하다는 주장에는 동의할 수 없다. 왜냐하면 본질적으로 내재한 프로그래머의 능력을 측정할 수 있는 현실적인 방법은 없기 때문이다.

프로그래밍 경진대회는 컴퓨터 과학의 제반요소를 가르치고 평가하는데 매우 중요한 한 방법 중의 하나라고 필자는 생각한다. 비록 전부는 아닐지라도 이를 적절히 활용한다면 큰 교육적 효과를 거둘 수 있다. 이런 취지로 국내외적으로 많은 경진대회가 운영되고 있다. 경진대회는 크게 공모형과 경시형이 있는데 공모형은 제시된 주제에 맞춰 완성된 결과물을 제출하고 그 내용을 평가받는 것인데 비하여 경시형은 비공개된 문제를 시합당일 제한된 장소와 조건에서 바로 풀고 그 결과로 평가는 받는다. 경시형과는 다른 공모형 프로그램 대회에 대한 연구도 있다[3].

국제규모급의 경시형 대회가 언제부터였는지는 확실하지는 않지만 1990년 USENET에서 인터넷으로 진행된 것이 최초라고 알려져 있다[6]. 이때는 특정 주관 기관이 있었던 것이 아니라 동호인(주로 교수나 연구소 연구원) 중심의 대회로 시작되었다. 최초대회에 참가한 60팀은 이후 계속된 3번의 대회에서 점점 늘어나 마지막 USENET 대회인 1992년도에는 약 14개국에서 출전한 290개 팀이 참가하여 3시간 동안 900개 정도의 프로그램을 제출하였다.

* 정회원

표 1 국제규모의 프로그래밍 대회

대회이름	주최	참가대상	진행방식
IOI	UNESCO	중고생	경시형
ICPC	ACM	대학생	경시형
ICFP	주최는 매년 공모	제약 없음	경시형
Imagine Cup	Microsoft	제약 없음	공모전
IEEE Extreme24	IEEE	대학생 회원	경시형

현재 국제적 규모의 프로그래밍 대회는 아래 표 1에 나열된 대회가 잘 알려져 있다. IOI는 대학생 이전의 중고등학생 영재 대상의 대회인데 주최는 비영리 UNESCO가 주관하고 있다. 그리고 ICFP는 Functional Language로 프로그래밍을 하는 대회이고 IEEE Extreme 24는 인터넷으로 24시간 동안 제약 없이 시행하는 대회이다. Microsoft에 주최하는 Imagine Cup은 공모전이기 때문에 평가기준의 주관성과 주최 측의 의지가 반영되기 때문에 어떤 정도의 내용이 수상을 하는지에 대한 기준이 모호한 점이 있다. 이는 공모전이 가진 공통적인 문제이다. 이중에서 가장 많이 알려지고 참가율이 높은 것은 단연 ICPC와 IOI이라고 할 수 있다.

2. 학생을 위한 프로그래밍 대회

현실에서 우리가 해결해야하는 문제의 주변상황은 항상 제한적일 수밖에 없다. 전형적인 예는 완성까지의 시간제한이다. 모든 과제에는 마감시간이 있고, 프로젝트에서 마감시간이 있다. 따라서 주어진 시간 안에서 최선의 결과를 만들어 내는 일은 언제나 중요한다. 문제는 그 제한시간의 정도에 따라서 다른 전략을 구사해야한다는 것인데, 일반적인 프로그래밍 과제와 같이 일(day) 단위부터 대형 과제의 단위인 월(month) 단위가 있다. 그러나 대부분 프로그래밍 대회는 대략 3-5시간 정도의 시간제한을 두고 있다.

그 다음 제한은 프로그래밍 자원에 따른 제한이다. 예를 들어 주어진 주 메모리, 또는 디스크 공간이 얼마 이하이여야만 되어야 한다는 것. 또는 수행시간은 반드시 일정 이하가 되어야 한다는 것을 들 수 있다. 최근 들어 일반 컴퓨터 시스템이 아닌 Embedded System에서의 프로그래밍은 이와 같은 전형적인 제한환경을 고려한 프로그래밍을 고려해야 한다. 또 다른 제한은 참가형태에 관한 제한이다. 과제에 따라서는 모든 팀원이 내용을 숙지한 상황에서 진행을 해야 할 필요가 있다. 그 이유에는 이후 관리(maintenance)의 문제를 해결하기 위한 것일 수도 있고, 전체 일에 대한 내부 조직원들의 이해도를 일정이상 올리기 위한 것일 수도 있다. 따라서 거의 모든 프로그래밍 과제는 정도의 차이는 있겠지만 항상 특정한 제한이 있기 마련이다. 이런 이유로 우리는 제한된 환경에서 프로그래밍 하는 연습을 학부생 때부터 적절히 할 필요가 있다. 필자는 학부에서 고급 프로그래밍(요즘은 문제해결기법으로 개명), 자료구조 등의 수업을 해왔다. 이 교과목을 진행하면서 느낀 바는 프로그래밍 과제에 대한 조건이 좀 더 제한적일수록 학생들의 참여도는 높았다고 판단한다.

필자는 좀 더 제한된 환경에서의 프로그래밍 과제를 〈컴퓨터 그래픽스〉에서 활용했다. 먼저 2시간 이론수업에서는 기본적인 내용과 이후 해야 할 과제의 내용에 대하여 설명을 한다. 그 다음 2시간동안 학생들은 필자가 준 과제를 반드시 마쳐야만 한다. 왜냐하면 담당조교가 마지막 10분부터 과제 검사를 하고, 주어진 2시간 이후에 제출한 과제는 모두 0점 처리를 하기 때문이다. 물론 학생들 간의 간단한 토론은 얼마든지 허용했지만, 코드를 주고받는 일은 일체 허용하지 않았다. 학생들의 반응은 다양했는데 대부분 긍정적이었다. 좀 고되긴 하지만 일단 과제를 마치고 나면 더 이상 집에 가서 까지 고민할 숙제가 없어 편했다는 의견, 집중이 잘 된다. 하루 반나절에 한 꼭지의 강의가 완성되어 좋았다는 등의 반응이 있었고, 프로그램에 익숙지 못한 학생들은 진도가 너무 빠르다, 차근차근 기초를 익힐 기회가 없다는 불만이 있었다.

물론 이 과정에 사용하는 과제물은 강사와 조교가 일차적으로 완성한 내용으로부터 시작한다. 필자가 가장 신봉하는 교육방법론은 Learn By Example이다. 이는 예제 코드를 제시하고 이를 실행하여 이해한 뒤에, 강사가 요구한 내용대로 고치고 추가하여 최종의 결과물을 완성하는 식의 프로그래밍 교육은 이전의 다른 방법에 비해서 훨씬 효과적이고 효율적이었다. 필자가 사용하는 제한된 환경에서의 프로그래밍 과제 수행에 관한 다른 연구자들의 결과도 있는데 이들이 주장하는 "Just-In-Time" 교육에 대한 진행상황과 학생들의 반응, 그리고 장단점에 대한 연구도 있다[2].

일반 과제, 즉 한 실험실에 수강생이 모두 들어가서 1인 작업을 할 수 없는 문제해결 기법, 자료구조 등의 과제물은 제시기간부터 마감기간까지의 시간을 아주 짧게 설정했다. 예를 들어 오늘 오후에 나눠준 과제물의 완성시점을 내일 아침이나 늦어도 내일 오후, 어떤 경우는 당일 저녁 12시까지 제출토록 제한을 하였다. 필자는 학생들의 과제를 Web 시스템으로

받고 바로 자동 채점을 하여 학생에게 결과를 돌려준다. 그리고 과제가 마감되고 나면 학생들 채점결과가 바로 Web에 자동으로 고시된다.

이 방식의 단점은 개발에 수주일 이상 걸리는 큰 프로그램은 과제로 줄 수 없다는 것인데, 이런 경우는 각 모듈별로 잘라서 잘게 나누어 주고 최종적으로 그 전체를 결합하여 제출하게 함으로서 어느 정도 극복할 수 있었다. 이런 식의 속전속결형 과제 제시와 평가를 이용하면 한 과목 강의에 많게는 18개까지의 프로그래밍 과제를 제시할 수 있었다.

학생들의 반응은 대체로 긍정적이었는데 비하여 잘 따라오지 못하는 학생은 상당수가 수강취소를 해서, 담당 강사로서 좀 마음이 아팠다. 재미있는 것은 거의 같은 내용의 과제물을 제시하고 환경을 느슨하게(허용시간을 하루에서 일주일) 했을 때 도리어 참여도나 이해도가 떨어진다는 좀 모순적인 결과였다. 그도 그럴 것이 학생들은 필자의 과목만을 수강하는 것은 아니기 때문에 과제의 대한 이해도는 시간이 갈수록 떨어지게 마련이다. 학생들 사이에 회자되는 "모든 과제는 마감 이틀 전부터 시작된다, Term Project 역시도" - 이 말은 이러한 상황을 적절하게 묘사하고 있다고 본다.

ICPC 대회와 같이 프로그래밍을 매우 제한된 상황에서 하도록 요구하는 교육의 장점과 단점을 나열하면 다음과 같다.

- 장점

(a) 프로그래밍 실력이 우월한 학생들은 다른 교육평가 방법에 비해서 매우 높은 성취감을 느낀다(차이를 정량적으로 느낄 수 있다).
(b) 중상위 그룹의 학생들의 참여도는 높다.
(c) 학생들의 집중도가 높아진다.
(d) 프로그래밍 환경이 제한될수록 학생들은 주체적으로 전략을 세우고 실행한다.
(e) 자신의 프로그래밍 능력을 빠른 시간에 정량적으로 확연하게 파악할 수 있다.

- 단점

(a) 교과목에 대한 선행학습이 부족한 학생은 쉽게 좌절감을 느낀다.
(b) 하위 그룹의 참여도는 일반 환경에서의 과제물 제시 때와 비교해서 떨어진다.
(c) 수업진행시 지나친 긴장감을 유발시켜 차분하게 문제의 근본을 생각하는 훈련에는 다소 방해가 될 수 있다. 어떤 학생들에게서는 마감시간이 다가올수록 상당한 강박 증상까지 보임을 관찰할 수 있었는데 이는 다른 조건을 완화시키는 방법으로 좀 더 개선시킬 여지가 있다[2].
(d) 참가학생들 지나친 경쟁 구도로 몰아가는 것은 자연스런 사회성에 기초한 토론과 협동적(collaborative) 탐구에 방해가 된다.
(e) 결과위주의 프로그램에 집중하므로 프로그램의 스타일이 조악해지기 쉬우며 전체적인 구조나 알고리즘의 개선에 집중할 수 있는 여유를 가지지 못한다.

Extreme Programming의 한 가지 방법인 Pair Programming[1,8]과도 약간의 연관이 있지만, 고도의 algorithmic solution이 필요한 ICPC에서 요구되는 상황과 복잡다단한 specification을 꼼꼼하게 다루는 Pair Programming적인 방법은 ICPC에서는 큰 도움이 되지 않을 듯하다.

요약하자면 프로그래밍 경시대회는 피교육자들의 효과적인 교육을 위해서도 매우 바람직한 한 가지 형태이다. 특히 학생들에게 프로그래밍은 단순히 typing speed의 문제 아니라 think speed의 문제라는 것을 확실하게 각인시켜주는 효과가 있다[6]. 또한 요즘 주목받고 있는 영재교육의 일환으로 재능 있는 고등학생들에게 일찍 프로그래밍의 세계에 접해주고자 하는 노력이 각국을 중심으로 일어나고 있는데, 이를 위해서도 프로그래밍 경시대회는 중요한 교육방법이 됨을 여러 연구는 밝히고 있다[5].

3. 교육자를 위한 ICPC 대회

국민 총소득 지수가 한 국가의 경제력의 정도를 암시해주듯이, ICPC 대회의 결과는 여러 의미를 지닌다. 겉으로 드러난 ICPC 성적은 각 국가의 최상위 대학생들의 프로그래밍, 알고리즘 구상력을 나타내준다고 해도 크게 틀리지 않다. 예를 들어 CACM에 기고된 한 글에서 미국의 전산학자는 ICPC 대회에서 갈수록 미국의 성적이 떨어지고 있는 상황을 매우 우려스럽게 보고 있다[7]. 심지어 그는 이렇게 탄식을 하고 있다.

"ICPC 대회에서 우승한 러시아 페테르부르크 대학생들은 모두 크렘린 궁전에 초대되어 러시아 대통령인 푸틴의 융숭한 칭찬과 환대를 받았는데 미국은 전 1996, 1997년에 미국이 우승할 당시 그 우승 대학 팀에게 도대체 어떤 대접을 하였는지 생각해 보자. 그들이 백악관 대통령을 만나기 위해서는 그 앞의 수많은 미식축구 우승 대학팀과 대통령과의 파티가 끝나기만을 기다려야 한다. 이런 이유 때문인가? 미국은 앞으로도 절대 미식축구(?)에서는 결코 세계 최고에서 밀

려나지 않을 것이다. 프로그래밍에 재능 있는 학생들을 지속적으로 찾아내서 교육하고 우대하지 않는다면 미국의 IT산업의 미래는 걱정스러울 수밖에 없다"[7].

최근에는 러시아와 중국, 청화대학을 중심으로 한 구 사회주의권 나라의 성적이 압도적이다. 특히 명문대학으로 급부상하고 있는 상하이 교통대학, 중산대학 등의 약진은 놀라울 정도이다. 물론 ICPC 성적이 국가, 좁게는 컴퓨터 전공 대학생들의 실력으로 바로 환치되는 것에는 무리가 있지만, 전혀 무관하다고 말할 수는 없을 것이다. 그것은 우수한 IT전공 학생들을 국가나 대학에서 얼마나 체계적으로, 지속적으로 관리하고 키워나가고 있는가 하는 총체적인 결합력을 나타내는 중요한 지표가 되기 때문이고, 이러한 총체적인 지원체제는 결국 국가의 경쟁력으로 언젠가는 나타나게 되어 있다.

특히 이론적인 내용을 학생들에게 재미있게 강의하기란 참으로 어려운 일이다. 그리고 매우 일반화된 강의에서 이를 실행하기라 쉽지 않다. 이 경우 프로그래밍 경시대회는 매우 효율적인 수단이 될 수 있음을 지적한 연구가 있다[4]. 예를 들어 NP-Complete problem(예를 들면 3-SAT이나 TSP 문제)의 성능에 관한 여러 대회에서 제시된 알고리즘은 매우 우수한 성과물로 확인된다. 마찬가지로 경쟁적인 환경을 통하여 학생들에게 이론적 문제의 구조와 어려움을 이해시키는 것은 교육적으로도 매우 좋은 방편이다. 본 특집호의 다른 글에서 자세히 소개가 되겠지만 ICPC에서 제출된 문제, 또는 고안되었지만 실제 대회에서 사용되지 못한 문제는 매우 훌륭한 프로그래밍 도전과제로 수업에서 사용될 수 있다.

필자가 ICPC 출제위원과 운영위원으로 참가해본 과정에서 새롭게 인식한 내용은 이후 일반 강의에도 매우 유익하였다. 보통 출제를 하게 되면 이 문제를 풀 수 있는 학생들이 몇 팀이나 되는지 그 푸는 시간이 얼마나 될지를 가늠하여 문제를 배치하게 되는데, 이 예측과 결과를 비교하면 많은 교육적인 사실을 배울 수 있다. ICPC 대회를 진행하면서 우리 학생들의 프로그래밍 습성이나 능력에 대하여 새롭게 알게 된 사실을 나열하면 다음과 같다.

(a) 우리나라 학생들은 협업적인 프로그래밍 작업에 매우 취약하다. 학생들은 주어진 문제가 9개라면, 참가자(3명)가 한번 읽어 본 뒤에 각 3문제씩 골고루 배분하는 경향이 있다. 개별 돌파를 하는 것이다. 이 경우 문제가 각자의 취향에 부합되거나 많이 다루어본 경우라면 매우 효율적인 전략이 되지만 한 사람만의 지식만으로 불가할 경우에는 아주 나쁜 결과를 초래한다. 이에 비하여 상위권 외국 학생들은 토론에 매우 익숙하여 시간이 지날수록 그 푸는 속도가 빨라진다.

이는 고등학교를 거쳐 대학에 와서도 제대로 된 토론형 교육을 받지 못한 이유 때문으로 보인다. 필자 역시 이 범주에서 크게 벗어나지 못한다. ICPC는 협업형 작업, 토론의 실제 문제 풀이에서 얼마나 도움이 되는지, 어떻게 토론을 하고 협동을 해야 하는지를 가르쳐 주는데 매우 좋은 훈련과정이 될 수 있다[1].

(b) 학생들은 test case 설계로 잘 훈련되어있지 못한 것으로 보인다. 경시용 문제를 위한 test case는 매우 다양한 경우를 담고 있기 때문에 세세한 부분까지 매우 꼼꼼하게 구현해야 한다. 왜냐하면 ICPC 채점에서는 준비한 test case를 모두 통과하면 비로소 정답으로 인정되기 때문에 부분점수(partial credit)가 없다. 이런 면에서 본다면 IOI 채점과는 완전히 다르다. IOI는 각 test case별로 부분점수를 준다. 이런 다양한 test case를 만드는 것은 채점 위원들에게도 도움 되는 경험이었다.

이를 위해서 필자가 사용한 방법 한 가지를 소개하고자 한다. 이전 IOI 출전을 위한 대표학생 교육에서 사용한 방식인데, 학생을 두 그룹으로 나눠서 한 쪽은 문제의 해답 프로그램을 작성하게 하고 다른 한 쪽은 그 문제에 사용될 test case를 설계하게 하였다. test case를 설계하는 쪽은 문제가 허용하는 범위 내에서 최대한 상대편 쪽 학생들이 통과되지 못하도록 worst case나 기묘한 경우를 고안하게 된다. 이러한 경험이 나중에 다시 양쪽의 역할을 바꾸어 문제를 풀게 하였을 때 큰 도움이 됨을 발견할 수 있었다. 일반 대학생들의 프로그래밍 실습의 경우에는 무작위로 2명씩 짝을 지어 한 사람이 다른 사람의 프로그램을 test하게 하였는데 이 역시 학생들의 흥미도 유발할 수 있었고 과제에 대한 집중도를 높이기에도 효과가 있었다.

(c) 내용에 대한 설명이 복잡한 문제의 경우에 기대한 정답률보다 낮았다. 이는 물론 문제의 설명이 영어(ICPC 아시아 지역예선의 공식 언어)로 되어 있다는 사실에도 기인하지만 그를 감안한다고 해도 기대보다는 낮았다. 복잡한 문제의 specification을 차근차근 분석적으로 이해하는 교육이 더 강화되어야 할 것이다.

(d) Dynamic Programming적인 해답이 요구되는 문제에 대해서는 상당히 익숙해져 있음을 볼 수 있었다. 사실 Dynamic Programming은 이해하기가 어려운 해결기법인데 이미 많은 대회를 통하여 몇 문제는 반드시(?) 이 형식으로 출제된다는 믿음이 있어서 그런지

기대보다 잘 풀었다.

(e) 계산 기하학(Computational Geometry) 관련 문제에 대한 학생들의 이해도나 정답률은 매우 낮았다. 계산 기하학 문제는 ICPC 대회에서 항상 주요 이슈가 된다. 왜냐하면 대부분의 학부에서 이 주제를 집중적으로 강의하지는 않기 때문이다. 알고리즘 과목의 뒷부분 기타 토픽에서 잠시 다루거나 아예 다루지 않거나 한다. 필자 역시 학부에서는 한 두 시간의 소개하는 강의 정도로 끝을 낸다.

프랑스의 주요 이공학과에서는 기하학을 매우 중요하게 여긴다. 프랑스에서 학위를 하신 학과 교수 한 분의 소개로 본 프랑스 교과목에는 기하학 관련과목이 학부에 무려 4과목이나 있어 상당히 놀란 적이 있다. 특히 최근 새롭게 대두되고 있는 Mobile Computing이나 Ubiquitous Sensor Network(USN)의 많은 문제가 계산기하학 문제로 수렴함을 볼 때 이제는 학부에서도 체계적으로 가르쳐야 할 때가 되지 않았는가 하는 생각을 해본다.

(f) 개별 컴퓨터에서 구현된 알고리즘의 실제적인 성능측정(instrumentation)에 관한 교육이 잘 되어있지 않는 듯하다. 예를 들면 이 정도의 계산량으로 looping을 수행하면 어느 정도의 시간이 소요되는지, 실제 어느 정도의 입력 데이터 크기에서 bubble sort가 확실히 느리게 되는지, Standard Template Library class를 쓰면 실제 수행속도가 빨라지는지, dynamic memory allocation을 쓰면 얼마나 느려지는지를 꼼꼼하게 millisecond 단위의 내부 clocking을 해가면서 측정해보지 않으면 실제 대회에서는 수행시간에 대한 가름을 할 수 없을 것이다. 이러한 알고리즘 공학론적 관점에서의 구체적인 성능측정에 관한 연습이 과제물이나 실험실습에서 강조되어야 할 것이다.

5. 글을 마치며

ACM-ICPC 대회는 프로그래밍 경시를 목적으로 하는 여러 국제대회 중에서 가장 인지도가 높은 국제규모의 행사이다. 우리나라에서는 이 대회의 준결승격인 아시아 지역예선이 매년 열리며 그 참가도와 열기는 갈수록 높아지고 있다. 이 ICPC 대회는 대학생들의 창의성 있는 알고리즘 설계를 위한 교육의 장이기도 하지만 이에 참가하는 교육자들을 위해서도 매우 좋은 기회를 제공한다.

ICPC는 창의적 알고리즘 설계에 재능 있는 대학생을 발굴하는 목적도 있지만, 이 과정을 통하여 우리나라 IT전공 대학생들의 수준을 진단하고 이를 개선하기 위한 방향을 모색하는 과정에서도 큰 도움이 된다. 또한 이 대회를 준비하는 여러 교육자들의 교육방향을 반성적으로 살펴보고 좀 더 효과적인 교육방법론을 개선하고 재구성하는 일에도 큰 도움이 된다.

진행상의 한 가지 생각해보아야 할 문제는 동아시아 예선 대회에서 상위권 몇 대학이 결선에 나갈 수 있는 상을 독점하는 경향으로 일반 중하위권 대학의 참여를 독려하기가 쉽지 않다는 것이다. 이를 개선하기 위한 여러 방책이 생각하여 시행을 해 보았다. 어떤 경우에는 ICPC 코치 자격의 교수가 운용하는 수업의 한 과제물을 ICPC대회 인터넷 예선참가로 대치하기도 하는데 좋은 방안으로 생각된다. 좀 더 독려한다면 ICPC 인터넷 예선에서 일정 이상의 성적을 받으면, 관련 수업의 성적에 가점을 해주는 경우인데, ICPC가 3인 1조의 단체전에는 각 3명에게 같은 가점이나 인센티브를 주어야하는 시행상의 문제도 생각해 봄직하다.

지금까지 ICPC 동아시아 예선은 컴퓨터이론 연구회를 중심으로 한 여러 참여교수님들의 헌신적인 노력 덕택에 어느 지역예선보다도 훌륭하게 진행되어 왔다. 지금까지의 노력을 종합하고 좀 더 체계적으로 정리하여 ICPC 아시아 지역예선이 한 단계 더 도약하는 계기가 본 특집호의 발간으로 앞당겨지기를 바라며 글을 맺는다.

참고문헌

[1] C. Macdowell et al, "The effects of pair-programming on performance in an introductory programming course", Proc. Software Engineering, ACM SIGCSE Bulletin archive, Vol. 34, Issue 1, pp. 38-42, Mar. 2002.

[2] O.Astrachan, "Non-competitive programming contest problems as the basis for just-in-time teaching," Proc. of the 34th Annual Frontiers in Education, 2004.

[3] S. Fitzgerald and M. Hines, "The computer science fair: an alternative to the computer programming contest," Proc. of SIGCSE, Philadelpia, pp.368-372, 1996.

[4] N. Shilov and K. Yi, "Engaging students with theory through ACM collegiate programming contest," Communications of the ACM, Vol.45, No.9, 2002.

[5] D. Myers and L. Null, "Design and implementation of a programming contest for high school students," ACM SIGCSE Bulletin, 1986.

[6] V. Khera, O. Astrachan and D. Kotz, "The internet programming contest: a report and philosophy," Proc. of the 24th SIGCSE technical symposium, 1993.
[7] D. A. Perterson, "Reflections on a Programming Olympiad," Communications on ACM, Vol.48, No.2 pp.7-8. 2005.
[8] http://xper.org/wiki/xp/PairProgramming

조 환 규

1984 서울대학교 계산통계학과(학사)
1986 한국과학기술원 전산학과(석사)
1990 한국과학기술원 전산학과(박사)
1990~현재 부산대학교 공과대학
2002 국제정보올림피아드(IOI) 출제위원장
1995~ 한국정보올림피아드 운영위원
관심분야 : 생물정보학, 이론전산학, 과학사회학
E-mail : hgcho@pusan.ac.kr