

적당히 틀린 답 찾기

경영학과 이찬우

컴퓨터의 역사는 여러 가지 한계를 극복하면서 걸어온 과정으로 볼 수 있다. 컴퓨터로 복잡한 문제를 해결하기 위한 한계들을 보자. 우선 튜링 기계로 자동적인 문제해결이 가능하다는 것을 알았다. 하지만 현실 속의 문제들을 튜링 기계로 표현하기에는 너무 복잡했다. 그래서 컴퓨터 과학자들은 좀 더 인간에게 가깝고 알고리즘 속의 문제를 쉽게 파악할 수 있는 언어 구조를 고안했다. 이 언어들은 튜링 기계 또는 람다 계산법의 뿌리에서 한 단계씩 올라와서 사람들이 문제를 쉽게 구조화할 수 있도록 표현법을 개선했다. 여기서 하나의 난관이 더 있다. 복잡한 문제를 해결하기 위해서는 입력값이 많은 문제를 해결해야 한다. 그러기 위해서는 기본적으로 많은 스위치가 필요하다.

부울 대수는 많은 입력값을 처리하기 위한 유용한 도구가 되었다. 컴퓨터는 많은 스위치를 논리에 맞게 배치할 수 없다면 거의 할 수 있는 일이 없다. 많은 스위치를 원하는 방향으로 구조화시킬 수 있는 방법은 간단하다. 우선 작은 스위치의 부품들을 하나하나 만든다. 그리고 작은 부품들을 쌓아나가는 것이다. 내가 사용하는 컴퓨터를 겉모습만 보면 속이 얼마나 복잡한지 모른다. 칩 속의 많은 스위치들을 복잡하게 배열하는 일은 정말 어려워 보인다. 하지만 단순한 부품들을 역할과 성질에 따라 조립하면 일이 훨씬 간단해진다. 시작은 미약한 스위치 조합이었지만 끝은 창대한 보편만능의 튜링 기계가 된다. 튜링의 머릿 속에만 존재하던 기계를 실제로 구현하기 위해 여러 제약들을 하나하나 해결하며 지금의 컴퓨터가 된 것이다.

이러한 컴퓨터의 발전 방향이 소프트웨어를 짜는 과정과 비슷하다는 생각이 들었다. 둘 다 뜬금없는 도약이 없다. 차근차근히, 작게 쪼갠 것을 뭉쳐서 만드는 소프트웨어처럼 컴퓨터의 역사도 작은 발걸음이 이끌어왔다. 단기간에 놀랍도록 성장한 컴퓨터의 발전 과정에 발판없는 도약은 없었다. 그저 차근차근, 쪼개서 붙이는 단순한 작업의 연속으로 다음 단계로 갈 수 있는 발판을 만들었다. 작은 튜링기계의 심볼 하나가 수백만 페이지에 달하는 소프트웨어로 성장하는 과정은 무엇인가? 작은 부품으로 나누고 더할 뿐이다. 간단하다. 복잡할 것은 없다. 소프트웨어를 만들려면 인내심 있게 목표만을 바라보고 묵묵히 다가가면 된다. 컴퓨터도 공학자, 수학자, 논리학자들이 조금씩 쌓아올린 탑 위에 있을 뿐이다.

컴퓨터를 보며 내가 성공에 관해 잘못된 편견을 가지고 있다는 것을 깨달았다. 나는 성공하기 위해서는 한번의 도약으로 역량을 끌어올려야 한다고 생각했다. 대학 입시라는 큰 관문에서의 성공, 고시나 변호사 시

힘을 통과했다는 자격증을 따면 한 순간에 성공할 수 있다고 생각했다. 하지만 인생이라는 프로그램에 비약이 존재할 수 있는지에 대해 의문이 들었다. 논리회로의 과정을 보면 인생에 비약이라는 것은 없을 지도 모르겠다고 생각했다. 성공한 사람들의 인생도 마찬가지다. 속내용을 감추면서 쌓아가기 때문에 그렇게 보일 뿐이다. 그 속을 들여다 보면 작은 노력들이 칩 속의 스위치처럼 잘게 쌓여있을 것이다. 결국 내가 목표로 하는 답은 지금 이 순간의 작은 부품이 모여서 만들어진다.

사람들은 많은 천재, 위인들은 한 건의 업적으로 명성을 얻은 것처럼 말한다. 이순신 장군의 명량 해전, 스티브 잡스의 아이폰 개발처럼 가장 빛나는 업적에만 초점을 맞춘다. 하지만 어릴 적 위인전을 읽을 때를 기억해보았다. 두꺼운 위인전 속에 멋진 업적을 이루는 스토리는 고작 몇 페이지에 불과하다. 나머지는 지루한 반복의 연속이었다. 어린 내가 보기엔 그들의 삶의 나머지는 재미없는 작은 기초 부품들로 가득 차 있었다. 그래서 위인들이 한 순간에 성공했다고 착각했다. 그들이 이룬 업적의 세부적인 부분은 재미없고 하찮은 부품처럼 보였다. 하지만 이 부품들이 조금씩으로 쌓이면서 낙숫물이 돌을 뚫듯이 질적인 변화가 일어났던 것이다. 무언가를 쌓아올리고 싶으면 그것을 받쳐줄 작은 기초 부품부터 만들어내야 한다.

가치있는 일을 하기 위해서는 무엇을 해야하는지 이제는 조금 알겠다. 작은 부품에 주목하자. 소프트웨어 부품에서 하찮은 것은 없다. 논리회로의 과정에 있는 부품들은 모두 동등한 가치를 가진다. 하나라도 빠지면 논리 비약이 되어 오류가 나기 때문이다. 내가 나중에 활용할 수 있는 작은 부품부터 쌓아가보자. 작은 것부터 실천하자. 사소해 보이는 지금의 일이 인생 전체에서는 중요한 부품이 될 수도 있다. 내 인생이라는 프로그램이 정확히 어떤 결과값을 낼지는 누구도 정확히 모른다. 하지만 일상 생활에서의 작은 용기, 도전이 인생의 부품이 되어 결과값을 바꿀 가능성이 있다는 것은 느낄 수 있었다. 내 인생이라는 프로그램을 위해 가장 적당한 답을 찾는 과정은 빅 픽처속의 퍼즐 조각을 하나하나 맞추는 것과 같다고 생각한다.

나는 내 인생이라는 문제에 답을 내는 프로그램의 시작점에 계속 멈춰있었다. 이 프로그램은 너무 복잡하고 거대해서 답을 찾는 방법은 물론 답이 있는지조차 모르겠다. 이리저리 겉모습만 훑어보았지만 그 복잡한 프로그램을 도대체 어떻게 시작할지 감이 잡히지 않았다. 그래서 두려움 앞에 겁을 먹고 움츠려 들었다. 어디에선가 한번의 돌파구를 찾을 수 있는 운명같은 기회가 오지 않을까 생각하고 있었다. 기다리고 또 기다리는 시간의 연속이었다. 어디서부터 시작해야 할까? 나는 진로나 삶의 가치관을 결정하는 문제에서 방향하고 있었다.

모든 걸 준비된 상태로 시작하고 싶었기 때문에 아무것도 시작하지 못했다. 사실 나는 NP 문제를 풀려고 시도한 것이나 다름없었다. 모든 일에 완벽한 준비를 한다는 것은 현실 시간 안에서 불가능에 가깝다. 그래도

나는 최고의 조건으로. 남들에게 도움 안 받고 스스로 잘 해내갈 수 있게 완벽한 상태가 되고 싶다고 생각했다.

창업을 해보고 싶었지만 불안했다. 창업을 하려면 기업법, 기술과 IT 분야의 지식, 모든 분야의 현재 창업 아이템, 경제, 재무, 회계, 마케팅, 언어, 문서 작성 등 모든 능력을 총괄할 수 있는 충분한 준비가 되어야 한다고 생각했다. 완벽하지 않고 적당한 수준에 머무른다면 실패할 것 같았다. 준비되지 않은 일은 하고 싶지 않았다. 그래도 어떤 도전도 해보지 않았다. 하지만 이런 내 이상은 주어진 시간안에 해결할 수 없는 문제이다. NP 문제를 현실 시간 안에 해결하려고 했다. 그리고 현실 시간 안에 해결할 방법을 찾지 못하자 나만의 답 찾기를 포기했다.

그리고는 남들이 시키는 대로, 칭찬해주는 대로 다른 사람들이 만들어놓은 길을 헤메며 걸어왔다. '저 공모전을 할 바에, 저 분야를 공부할 바에 남들이 인정해주는 이 분야를 배우는 게 낫겠지. 다른 걸 배우는 건 시간낭비야. 이거 하기도 바쁜데 저걸 어떻게 해?' 이렇게 말하면서 열심히 걷고 걸었다. 그런데 아무리 가도 결국 내가 원하는 방향을 아니었다. 그 방향으로 보기를 스스로 거부하고 눈, 귀를 막고 걸어왔기 때문이다. 남들이 인정해주면 마음이 편하니까, 잘하고 있다고 안심할 수 있으니까.

그런데 컴퓨터 과학에서 내 인생 프로그램의 답을 찾는 방법에 대한 힌트를 얻었다. 삶을 꾸려나가는 방법을 찾는 문제는 NP 문제다. $P \neq NP$ 라면 슈퍼 컴퓨터로도 답을 찾을 수 없다. 그래서 컴퓨터 과학이 쓰는 지혜를 차용하기로 했다. 적당히라도 결과를 한 번 내보자. '적당한 답이 아주 틀렸을 때는 완전히 잘못된 해를 가질 수도 있지 않을까?'하는 두려움은 어떻게 극복할까. 적당히 찾은 답이 틀린 경우에도 답에는 더 가까워질 수 있다. '다른 걸로 해보면 이것보단 낫겠군.' 하면서 말이다. 전공수업에서 최적화 문제를 푸는 방법은 대개 틀린 초기해로부터 최적해를 찾아나가는 방법을 배웠다. 시행착오는 가장 단순하면서도 강력한 알고리즘이 아닐까? 어떤 일을 하든 처음부터 완전하고 실수 없는 해를 찾자는 집착을 버리면 더 나은 답을 찾을 수 있을지 모른다.