

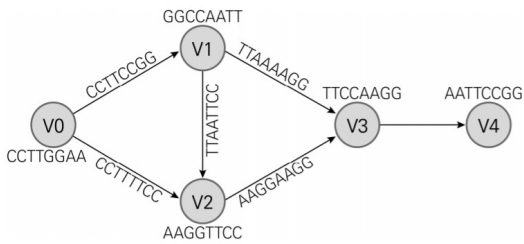
한 가닥의 DNA는 아데닌(A), 구아닌(G), 시토신(C), 티민(T)의 네 종류의 염기를 가지고 있는 뉴클레오티드가 선형적으로 이어진 사슬로 볼 수 있다. 보통의 경우 <그림 1>과 같이 두 가닥의 DNA가 염기들 간 수소 결합으로 서로 붙어 있는 상태로 존재하는데, 이를 ‘이중나선 구조’라 부른다. 이때 A는 T와, G는 C와 상보적으로 결합한다. 온도를 높이면 두 가닥 사이의 결합이 끊어져서 각각 한 가닥으로 된다.



<그림 1> 염기들 간 상보적 결합의 예

정보과학의 관점에서는 DNA도 정보를 표현하는 수단으로 볼 수 있다. 한 가닥의 DNA 염기서열을 4진 코드로 이루어진 특정 정보로 해석할 수 있기 때문이다. 즉, ‘A’, ‘G’, ‘C’, ‘T’만을 써서 순서가 정해진 연속된 n개의 빈칸을 채울 때, 총 4ⁿ개의 정보를 표현할 수 있고 이중 특정 연속체를 한 가지 정보로 해석할 수 있다.

DNA로 정보를 표현한 후, DNA 분자들 간 화학 반응을 이용하면 연산도 가능하다. 1994년 미군의 정보과학자 에이들먼은 『사이언스』에 DNA를 이용한 연산에 대한 논문을 발표했다. 이로써 ‘DNA 컴퓨팅’이라는 분야가 열리게 되었다. 이 논문에서 에이들먼이 해결한 것은 정점(예: 도시)과 간선(예: 도시 간 도로)으로 이루어진 그래프에서 시작 정점과 도착 정점이 주어졌을 때 모든 정점을 한 번씩만 지나는 경로를 찾는 문제, 즉 ‘해밀턴 경로 문제(HPP)’였다. HPP는 정점의 수가 많아질수록 가능한 경로의 수가 급격하게 증가하기 때문에 소위 ‘어려운 문제’에 속한다.



<그림 2> 정점 5개로 구성된 그래프

DNA 컴퓨팅의 기본 전략은, 주어진 문제를 DNA를 써서 나타내고 이를 이용한 화학 반응을 수행하여 답의 가능성이 있는 모든 후보를 생성한 후, 생화학적인 실험 기법을 사용하여 문제 조건을 만족하는 답을 찾아내는 것이다. 에이들먼이 HPP를 해결한 방법을 <그림 2>의 그래프를 통해 단순화하여 설명하면 다음과 같다. <그림 2>는 V0가 시작의 정점, V4가 도착 정점이고 화살표로 간선의 방향을 표시한 그래프를 보여 준다. 즉, V0에서 V1로는 갈 수 있으나 역방향으로는 갈 수 없다. 먼저 그래프의 각 정점을 8개의 염기로 이루어진 한 가닥 DNA 염기서열로 표현한다. 그리고 각 간선을 그 간선이 연결하는 정점의 염기서열로부터 취하여 표현한다. 즉, V0(<CCTTGGA>)에서 출발하여 V1(<GGCCAATT>)

에 도달하는 간선의 경우는 V0의 뒤쪽 절반과 V1의 앞쪽 절반을 이어 붙인 염기서열 <GGAAGGCC>의 상보적 코드 <CCTTCCGG>로 나타낸다. 이렇게 6개의 간선 각각을 DNA 코드로 표현한다.

이제 DNA 합성 기술을 사용하여 이들 코드를 종류별로 다량 합성한다. 이들을 하나의 시험관에 넣고 서로 반응을 시키면 DNA 가닥의 상보적 결합에 의한 이중나선이 형성되는데, 이것을 ‘혼성화 반응(hybridization)’이라 한다. 혼성화 반응의 결과로 경로, 즉 정점들의 연속체가 생성된다. 시험관 안에는 코드별로 막대한 수의 DNA 분자들이 있기 때문에, 이들 사이의 이러한 상호 작용은 대규모로 일어난다. ① 이상적인 실험을 가정한다면, 혼성화 반응을 통해 <그림 2> 그래프의 가능한 모든 경로에 대응하는 DNA 분자들이 생성된다. 경로의 예로 (V0, V1), (V1, V2), (V0, V1, V2) 등이 있다. 이와 같이 생성된 경로들로부터 해밀턴 경로를 찾아 나간다.

- [1단계] V0에서 시작하고 V4에서 끝나는지 검사한 후, 그렇지 않은 경로는 제거한다.
- [2단계] 경로에 포함된 정점의 개수가 5인지 검사한 후, 그렇지 않은 경로는 제거한다.
- [3단계] 경로에 모든 정점이 포함되었는지 검사한다.
- [4단계] 지금까지의 과정을 통해 취한 경로들이 문제에 대한 답이라고 결정한다.

에이들먼은 각 단계를 적절한 분자생물학 기법으로 구현했다. 그런데 DNA 분자들 간 화학 반응은 시험관 내에서 한꺼번에 순간적으로 일어난다는 특성을 갖고 있다. 요컨대 에이들먼은 기존 컴퓨터의 순차적 연산 방식과는 달리, 대규모 병렬 처리 방식을 통해 HPP의 해결 방법을 제시한 것이다. 이로써 DNA 컴퓨팅은 기존의 소프트웨어 알고리즘이나 하드웨어 기술로는 불가능했던 문제들의 해결에 대한 잠재적인 가능성을 보여주었다.

01

DNA 컴퓨팅에 대한 설명으로 적절하지 않은 것은?

정답 ⑤

- ① 창시자는 미국의 정보과학자 에이들먼이다.
에이들먼이 논문을 발표하면서 DNA 컴퓨팅이 창시되었다(3문단).
- ② DNA로 정보를 표현하고 이를 이용하여 연산을 하는 것이다.
DNA로 정보를 표현한 후, DNA 분자들 간 화학 반응을 이용하여 연산을 할 수 있다(3문단).
- ③ 기본적인 해법은 가능한 모든 경우의 수를 생성한 후, 여기서 답이 되는 것만을 찾아내는 것이다.
답의 가능성이 있는 모든 후보를 생성한 후, 조건을 만족하는 답을 찾아낸다(4문단).
- ④ 기존 컴퓨터 기술의 발달을 전환하여 분자생물학적인 방법으로 접근함으로써 정보 처리 방식의 개선을 모색했다.
분자생물학 기법을 통해, 기존 컴퓨터의 순차적 연산 방식 대신 대규모 병렬 처리 방식을 제시하였다(6문단).
- ⑤ DNA 컴퓨팅을 이용하여 HPP를 풀 때, 간선을 나타내는 DNA의 염기 개수는 정점을 나타내는 DNA의 염기 개수의 두 배다.
<그림 2>에서, 간선과 정점은 둘 다 8개의 염기로 구성된 DNA로 나타내진다.

02

㉠에 대한 설명으로 적절하지 않은 것은?

정답 ④

- ① (V1, V2, V3, V4)는 정점이 네 개이지만, 에이들먼의 해법 [1단계]에서 걸러진다.
(V1, V2, V3, V4)는 경로가 V1에서 시작하는데, V0에서 시작하지 않는 경로는 [1단계]에서 제거된다.
- ② V3에서 V4로 가는 간선으로 한 가닥의 DNA <TTCCTTAA>가 필요하다.
V3(<TCCAAGG>)와 V4(<AATCCGG>)에서 각각 뒤쪽 절반과 앞쪽 절반을 이어 붙이면 <AAGGAATT>가 되고, 간선은 이의 상보적 코드인데(4문단) A는 T와, G는 C와 상보적으로 결합하므로(1문단) 간선은 <TTCCTTAA>이다.
- ③ 정점을 두 개 이상 포함하는 경로는 두 가닥 DNA로 나타내어진다.
경로는 혼성화 반응에 의해 형성되는데(5문단), 혼성화 반응은 DNA 가닥의 상보적 결합에 의해 이중나선이 형성되는 것이다. 이때 이중나선 구조는 두 가닥의 DNA가 결합한 것이므로(1문단), 정점을 두 개 이상 포함하는 경로는 두 가닥 DNA로 나타내어진다.
- ④ 정점을 세 개 포함하고 있는 경로는 모두 네 개이다.
<그림 2>에서 살펴보면, (V0, V1, V2), (V0, V1, V3), (V0, V2, V3), (V1, V2, V3), (V1, V3, V4), (V2, V3, V4)로 모두 여섯 개이다.
- ⑤ 해밀턴 경로는 (V0, V1, V2, V3, V4) 뿐이다.
V1 직전에는 반드시 V0이 와야 하며, V1 이후에 V3으로 갈 경우 V2로 돌아갈 방법이 없다. 따라서 해밀턴 경로는 (V0, V1, V2, V3, V4) 뿐이다.

윗글과 <보기>를 비교 평가한 것으로 적절하지 않은 것은?

정답 ③

<보기>

DNA 컴퓨팅의 실용화를 위해서는 여러 기술적인 문제점들을 해결해야 한다. 그중 하나는 정보 처리의 정확도다. DNA 컴퓨팅은 화학 반응에 기반을 두는데, ㉠ 반응 과정상 오류가 발생할 경우 그릇된 연산을 수행하게 된다.

- ㄱ. ㉠가 발생하지 않는다면, <그림 2> 그래프에서는 에 이들면의 [3단계]가 불필요하다.
- ㄴ. 혼성화 반응에서 엉뚱한 분자들이 서로 붙는 것을 방지할 수 있도록 DNA 코드를 설계하는 것은 ㉠를 최소화하기 위한 방법이다.
- ㄷ. DNA 컴퓨팅의 원리를 적용한 소프트웨어를 개발하면, ㉠를 방지하면서도 대규모 병렬 처리를 통한 문제 해결이 기존 컴퓨터에서 가능하다.

- ① ㄱ ② ㄴ ③ ㄱ, ㄴ
- ④ ㄱ, ㄷ ⑤ ㄴ, ㄷ

ㄱ. [2단계]에서는 정점의 개수가 5가 아닌 경로를 제거한다. 만약 결합에 오류가 없다면, 경로에는 같은 정점이 두 번 이상 포함될 수 없다. 따라서 ㉠가 발생하지 않았다면, 정점의 개수가 5인 경로는 반드시 모든 정점을 포함하므로, [3단계]는 불필요하다.

ㄴ. 엉뚱한 분자들이 서로 붙도록 DNA 코드를 설계하면 당연히 오류가 발생할 것이다. 예를 들어, V0이 <CCTTGAA>인데 V4를 <CCTCCGG>로 설계할 경우, V0의 뒤쪽 절반이 V4의 뒤쪽 절반과 결합하게 되어 오류가 발생한다. <그림 2>에서, V0이 V4와 결합하는 경로는 불가능하기 때문이다. 따라서 이와 같은 오류를 방지하려면, 엉뚱한 분자들이 서로 붙지 않도록 DNA 코드를 설계해야 한다.

ㄷ. DNA 컴퓨팅의 원리가 소프트웨어에 적용된다고 해서, 오류가 방지된다고 볼 수 없다. 반응 과정상 오류는 분자생물학적 '방법'에서 기인하는 것이지, 분자 자체에서 기인하는 것이 아니기 때문이다. 즉 DNA 분자 대신 소프트웨어를 사용한다고 하더라도, 방법이 달라지지 않으면 오류는 방지되지 않는다.