Simulated Annealing

전역 최적화 문제에 대한 확률적 메타 휴리스틱

목차

냐냐냐

- Simulated Annealing(이하 SA)이란?
  - 왜 만들었지?
  - 알고리즘 대회에서의 이용
  - AI에서의 이용
  - SA 과정 이해하기
- SA으로 3-SAT 풀어보기
  - NP 클래스란?
  - 3-SAT이 갖는 의미
  - 백준 3-SAT 문제

왜 만들었지? - (1)

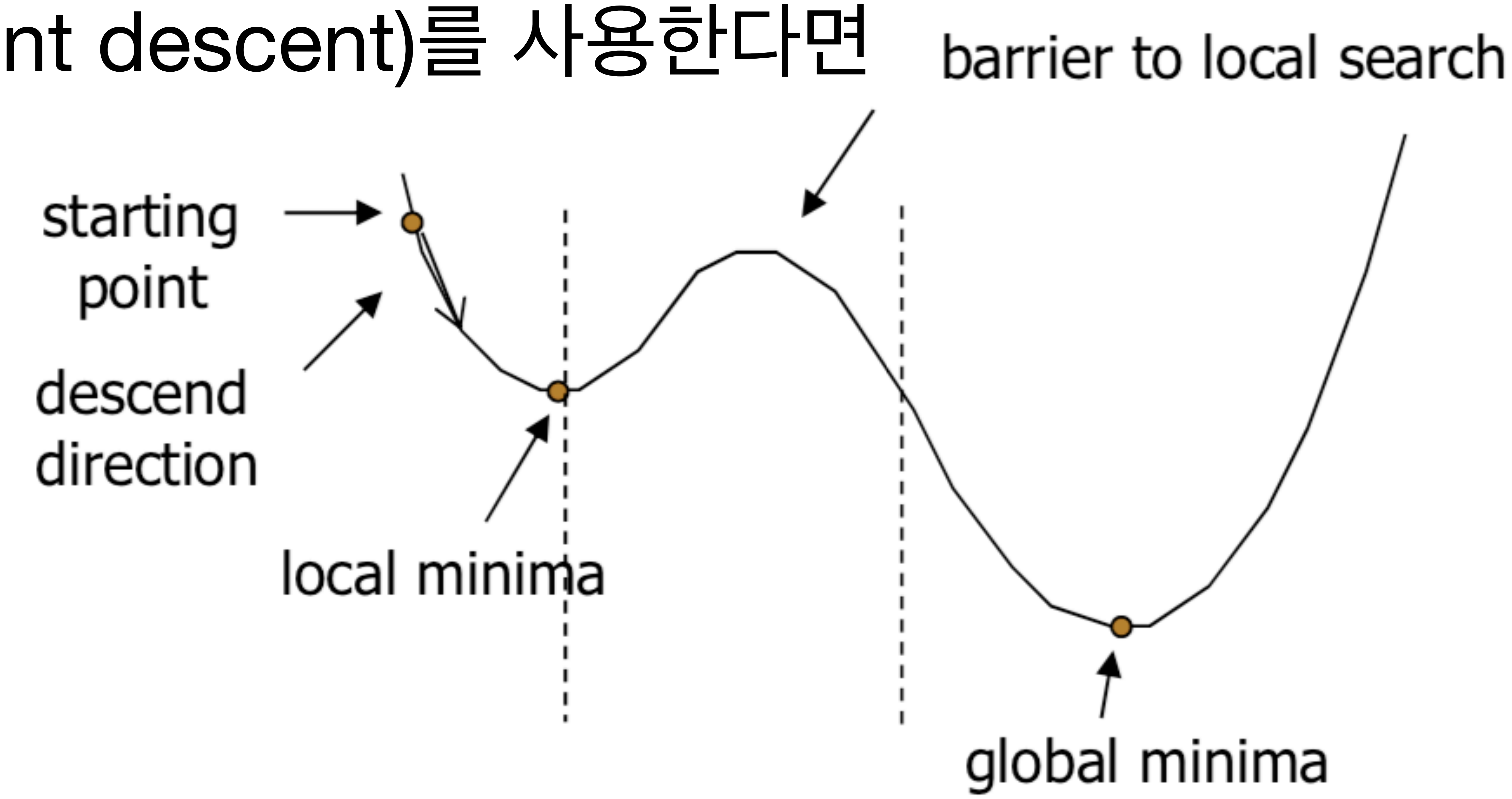
SA이란?

- 전역 최적화(Global optimization)
  - 되게 큰 탐색 공간에서 함수의 전역 최적점(Global optimum)의 근사치를 찾는 것
- SA는 전역 최적화에 일반적으로 사용할 수 있는 확률적 메타 휴리스틱
- 1983년 S.Kirkpatrick과 C.D. Gelatt, M.P. Vecchi와 1985년 V. Cerny가 각자 독자적으로 발명했다고 한다
- Simulated Annealing은 '담금질 기법'이란 오역으로 알려져 있다
  - '담금질'이 아니라 '풀림'이 맞다
  - 풀림은 금속재료를 가열한 다음 조금씩 냉각해 결정을 성장시켜 그 결함을 줄이는 작업이다
  - 실제로 SA의 정당성은 '열역학적 평형'을 통한 물리적인 비유로 이해할 수 있다 (뒤에서 마저 설명)

왜 만들었지? - (2)

SA이란?

- 지역 최적점(Local optimum)에 빠지게 되어도 적당히 움직여서 전역 최적점(Global optimum)으로 갈 수 있도록 도와주기 때문에 효과적이다
  - 따라서 지역 최적점이 많은 곤란한 상황에 사용하기 좋다
- 다음 상황을 가정했을 때, 경사 하강법(Gradient descent)를 사용한다면 barrier를 넘지 못하는 상황이 발생할 것이다



왜 만들었지? - (3)

SA이란?

- 폭넓은 응용 가능성과 최상에 가까운 해답을 낸다는 점에서 유용한 휴리스틱이라고 한다
- 하지만 좋은 해답을 얻는데 걸리는 계산 시간이 긴 것이 흠이라고 한다
  - 그러나 대규모 병렬 처리(Massively parallel execution)를 통해 어느 정도 완화할 수 있다고 한다

알고리즘 대회에서의 이용

SA이란?

- 일반적인 알고리즘 대회 문제들은 결정론적이다 (맞고 틀림이 분명함)
- 그럼에도 (아주) 종종 휴리스틱을 유도하는 문제들이 있는데, 이 때 SA가 자주 이용된다
- TMI) 삼성의 SCPC나 넥슨의 NYPC 같이 주로 기업 주관 대회에서 나온다

- 덤) 직접 짜보니까 구현은 매우 쉽다!

AI에서의 이용

SA이란?

- 사실 잘 모른다
- 경사 하강법이나 Hill climbing들이 AI에 잘 쓰인다고 아는데, 얘네의 발전된 버전인 SA도 잘 쓰이지 않을까?

SA 과정 이해하기 - (1)

SA이란?

- 담금질 할 때 **금속**이 vs SA 할 때 **알고리즘**이
    - 뜨거우면: 구조가 약하고 변화 쉬움 = 온도가 높으면: 정밀도가 낮고 변화 쉬움
    - 차가우면: 구조가 단단하고 변화 적음 = 온도가 낮으면: 정밀도가 높고 변화 적음
- 에너지 크기를 증가시킬 확률: $P(\Delta E, T) = e^{-\frac{\Delta E}{kT}}$ (by 맥스웰-볼츠만 분포)
- 이미지

SA 과정 이해하기 - (2)

SA이란?

- 인접 상태 정의하기
  - 변화가 최대한 작아야 한다!
  - 확률이 높은 인접 상태를 고른다면 더 좋다!
- 평가 함수 정의하기
  - 평가가 최대한 단순해야 한다!
- 이동하기

- 온도 감소시키기
  - 온도 감률 방식에는
    - 선형 감률($t \leftarrow t - \alpha$)
    - 기하 감률($t \leftarrow t \times \alpha$)
    - Slow-decrease 감률 ($t \leftarrow t/(1 + \alpha t)$)

NP 클래스란? - (1)

SA으로 3-SAT 풀어보기

- 알고리즘에는 복잡도(Complexity)가 있다
  - $O(n^k)$은 다항으로, 현실적인 비용이다
    - Why? 컴퓨터가 p배 빨라지면 $\times \sqrt[k]{p}$의 입력을 같은 시간에 처리할 수 있다
  - $O(k^n)$이나 그보다 심한 것들은, 비현실적으로 비효율적이다
    - Why? 컴퓨터가 p배 빨라지면 $+\log_k p$의 입력을 같은 시간에 처리할 수 있다
- 다항 정도로 풀리면 P 클래스다
- 운에 기댔을 대 현실적인 비용으로 풀리면 NP 클래스다
  - 다르게는, '채점을 현실적으로 할 수 있으면' 이라고도 한다
  - 덤) $P \subset NP$이고, $P - NP$문제는 $P = NP$인지를 묻는 세계 7대 밀레니엄 난제로 잘 알려져 있다

NP 클래스란? - (2)

SA으로 3-SAT 풀어보기

- NP 클래스에서 $NP-$완전은 모든 NP 문제로부터 환원 가능한 문제다
  - 즉, NP에서 제일 어려운 문제들이다
- $NP-$난해는 적어도 모든 NP 만큼은 어려운 문제들이다
  - NP와 $NP-$난해의 교집합에 있는 문제들은 $NP-$완전이다
- Stephen Cook은 1971년, 최초로 $NP-$완전에 포함되는 문제를 발견했다
  - Cook-Levin theorem은 SAT(Satisfiability problem)이 $NP-$완전임을 증명한다
  - 그 이후로도 다양한 $NP-$완전이 발견되어서 이제는 수천 개 가량의 $NP-$완전이 알려져 있다고 한다

3-SAT이 갖는 의미

SA으로 3-SAT 풀어보기

- CNF(Conjunctive normal form)는 리터럴들의 논리합(AND)로 이뤄진 절(Clause)들의 논리곱(OR)로 나타낸 논리식을 말한다
    - 모든 명제 논리식은 이중부정 법칙, 드모르간 법칙, 분배 법칙 등을 써서 전부 CNF로 변환 가능하다
- SAT(Satisfiability problem)은 CNF의 값을 참(True)로 만들기 위한 해를 구하는 문제이다
    - 모든 k-SAT는 3-SAT로 다항 시간에 환원할 수 있다
    - 따라서 3-SAT가 SAT를 대표하는 중요한 문제다
- 앞서 말했듯, 3-SAT은 NP 클래스에 속하는 문제다
    - 그래서 운에 기대는 비결정론적(Non-deterministic) 방법으로 풀어야 한다
- 덤) 2-SAT은 P 클래스에 속하고, Robert Tarjan이 SCC를 이용해 신박한 방식으로 해결했다

백준 3-SAT 문제

SA로 3-SAT 풀어보기

3-SAT

스페셜 저지 전체 채점

시간 제한	메모리 제한	제출	정답	맞힌 사람	정답 비율
2 초	512 MB	2029	386	46	4.299%

문제

3-SAT은 N개의 불리언 변수 $x_1, x_2, \ldots, x_n$가 있을 때, 3-CNF 식을 true로 만들기위해 x_i를 어떤 값으로 정해야하는지를 구하는 문제이다.

3-CNF식은 $(x \vee y \vee \neg z) \wedge (x \vee \neg y \vee z) \wedge (\neg w \vee x \vee \neg z) \wedge (x \vee z \vee y)$ 와 같은 형태이다. 여기서 괄호로 묶인 식을 절(clause)라고 하는데, 절은 3개의 변수를 ∨한 것으로 이루어져 있다. ∨는 OR, ∧는 AND, ¬은 NOT을 나타낸다.

변수의 개수 N과 절의 개수 M, 그리고 식 f가 주어졌을 때, 식 f를 true로 만들 수 있는지 없는지를 구하는 프로그램을 작성하시오.

예를 들어, N = 3, M = 4이고, $f = (\neg x_1 \vee x_2 \vee x_3) \wedge (\neg x_1 \vee \neg x_2 \vee x_3) \wedge (x_1 \vee \neg x_2 \vee x_3) \wedge (x_3 \vee \neg x_2 \vee \neg x_1)$ 인 경우에 x_1을 false, x_2을 false, x_3를 true로 정하면 식 f를 true로 만들 수 있다. 하지만, N = 1, M = 2이고, $f = (x_1 \vee x_1 \vee x_1) \wedge (\neg x_1 \vee \neg x_1 \vee \neg x_1)$인 경우에는 x_1에 어떤 값을 넣어도 식 f를 true로 만들 수 없다.

입력

첫째 줄에 변수의 개수 N (1 ≤ N ≤ 100)과 절의 개수 M (1 ≤ M ≤ 1000)이 주어진다. 둘째 줄부터 M개의 줄에는 절이 주어진다. 절은 세 정수 i, j, k (1 ≤ |i|, |j|, |k| ≤ N)로 이루어져 있으며, i, j, k가 양수인 경우에는 각각 x_i, x_j, x_k를 나타내고, 음수인 경우에는 $\neg x_{-i}$, $\neg x_{-j}$, $\neg x_{-k}$를 나타낸다.

출력

첫째 줄에 식 f를 true로 만들 수 있으면 1을, 없으면 0을 출력한다.

f를 true로 만들 수 있는 경우에는 둘째 줄에 식 f를 true로 만드는 x_i의 값을 x_1부터 순서대로 출력한다. true는 1, false는 0으로 출력한다.

백준 3-SAT 문제

SA로 3-SAT 풀어보기

40분간의 21번만의 제출만에 모든 테스트 케이스를 맞추는 쾌거를 이뤘다!!

52203753	dohoon	3-SAT	OK (204/204)	2144 KB	1892 ms	C++17 / 수정	1424 B	4분 전
52203750	dohoon	3-SAT	OK (202/204)	2144 KB	1864 ms	C++17 / 수정	1424 B	4분 전
52203745	dohoon	3-SAT	OK (203/204)	2144 KB	1824 ms	C++17 / 수정	1424 B	4분 전
52203738	dohoon	3-SAT	TLE			C++17 / 수정	1424 B	5분 전
52203706	dohoon	3-SAT	OK (203/204)	2144 KB	1768 ms	C++17 / 수정	1424 B	8분 전
52203702	dohoon	3-SAT	OK (203/204)	2144 KB	1748 ms	C++17 / 수정	1424 B	8분 전
52203698	dohoon	3-SAT	OK (203/204)	2144 KB	1772 ms	C++17 / 수정	1424 B	8분 전
52203677	dohoon	3-SAT	OK (201/204)	2144 KB	1384 ms	C++17 / 수정	1454 B	10분 전
52203663	dohoon	3-SAT	OK (202/204)	2144 KB	1760 ms	C++17 / 수정	1469 B	12분 전
52203659	dohoon	3-SAT	OK (202/204)	2144 KB	1784 ms	C++17 / 수정	1569 B	12분 전
52203650	dohoon	3-SAT	OK (203/204)	2144 KB	1784 ms	C++17 / 수정	1532 B	14분 전
52203630	dohoon	3-SAT	OK (203/204)	2144 KB	1960 ms	C++17 / 수정	1496 B	15분 전
52203622	dohoon	3-SAT	TLE			C++17 / 수정	1533 B	16분 전
52203605	dohoon	3-SAT	OK (202/204)	2144 KB	1928 ms	C++17 / 수정	1422 B	17분 전
52203579	dohoon	3-SAT	OK (202/204)	2144 KB	1600 ms	C++17 / 수정	1422 B	20분 전
52203476	dohoon	3-SAT	OK (49/204)	2144 KB	1588 ms	C++17 / 수정	1426 B	28분 전
52203469	dohoon	3-SAT	TLE			C++17 / 수정	1426 B	29분 전
52203445	dohoon	3-SAT	OK (50/204)	2144 KB	776 ms	C++17 / 수정	1488 B	31분 전
52203397	dohoon	3-SAT	OK (50/204)	2132 KB	752 ms	C++17 / 수정	1396 B	35분 전
52203362	dohoon	3-SAT	OK (45/204)	2132 KB	748 ms	C++17 / 수정	1380 B	38분 전
52203305	dohoon	3-SAT	OK (54/204)	2132 KB	496 ms	C++17 / 수정	1299 B	41분 전

백준 3-SAT 문제

SA로 3-SAT 풀어보기

21번의 시도면 꽤 빨리 맞춘 거져? ㅎㅎ 칭찬해주세염

그리고 20위부터는 203/204임 ㅎㅎ

1	12046374	40	august14	204/204	1256 KB	4 ms	C++14	4308 B	3년 전
2	32630129	44	ljjk159	204/204	2200 KB	12 ms	C++17	9679 B	1년 전
3	12080417	9	imeimi2000	204/204	2160 KB	1424 ms	C++17	4443 B	3년 전
4	33543387	39	jhnah917	204/204	2252 KB	1500 ms	C++14	4632 B	1년 전
5	37986428	1	zzammin	204/204	2252 KB	1500 ms	C++17	4632 B	10달 전
6	41781843	13	kerochuu	204/204	2252 KB	1500 ms	C++17	4632 B	7달 전
7	47332678	2	as32385	204/204	2252 KB	1500 ms	C++17	4632 B	3달 전
8	50417382	1	tikcher	204/204	2252 KB	1500 ms	C++17	4632 B	1달 전
9	44899788	1	sson18	204/204	2252 KB	1500 ms	C++17	4680 B	5달 전
10	37320415	1	MenOfPassion	204/204	2252 KB	1500 ms	C++17	4688 B	10달 전
11	12054585	79	njw1204	204/204	2004 KB	1604 ms	C++14	3986 B	3년 전
12	33569184	120	jinhan814	204/204	2116 KB	1640 ms	C++17	3035 B	1년 전
13	11994259	15	exqt	204/204	2000 KB	1812 ms	C++11	1168 B	3년 전
14	36463987	36	jthis	204/204	2128 KB	1828 ms	C++14	2059 B	11달 전
15	52203753	21	dohoon	204/204	2144 KB	1892 ms	C++17 / 수정	1424 B	7분 전
16	36205366	59	qwerasdfzxcl	204/204	2136 KB	1928 ms	C++17	2339 B	11달 전
17	12082431	37	leejseo	204/204	2120 KB	1968 ms	C++14	1471 B	3년 전
18	12046023	12	ainta	204/204	3964 KB	1976 ms	C++11	1281 B	3년 전
19	12093416	26	messi	204/204	2000 KB	1984 ms	C++11	949 B	3년 전
20	42550681	33	baba1	203/204	2124 KB	1896 ms	C++20	1082 B	7달 전

백준 3-SAT 문제

SA로 3-SAT 풀어보기

```cpp
#include <bits/stdc++.h>
#define rep(i,a,b) for (auto i = (a); i <= (b); ++i)
using namespace std;

const int N = 103, M = 1003;
int n, m, x[N], c[M][3], d[M][3];
mt19937_64 seed(314159);

int scoring() {
    int r = 0;
    rep(i,1,m) {
        bool C = false;
        C |= c[i][0] > 0 ? x[c[i][0]] : !x[-c[i][0]];
        C |= c[i][1] > 0 ? x[c[i][1]] : !x[-c[i][1]];
        C |= c[i][2] > 0 ? x[c[i][2]] : !x[-c[i][2]];
        r += C;
    }
    return r;
}

void simulated_annealing(double kt, double d) {
    uniform_int_distribution<int> I(1,n);
    uniform_real_distribution<double> R(0,1);
    double e0 = 0, e1, p;
    rep(_,1,74100) {
        int i = I(seed); x[i] ^= 1;
        e1 = scoring();
        p = exp((e1-e0)/kt);
        if (p < R(seed)) x[i] ^= 1;
        else e0 = e1;
        kt *= d;
        if (e1 == m) {
            cout << "1\n";
            rep(i,1,n) cout << x[i] << ' ';
            exit(0);
        }
    }
}

int main() {
    cin.tie(0)->sync_with_stdio(0);
    cin >> n >> m;
    rep(i,1,m) cin >> d[i][0] >> d[i][1] >> d[i][2];
    rep(_,1,10) {
        rep(i,1,m) rep(j,0,2) c[i][j] = d[i][j];
        fill(x+1,x+n+1,0);
        simulated_annealing(6.5,0.9999);
    }
    cout << 0;
}
```


```cpp
#include <bits/stdc++.h>
#define rep(i,a,b) for (auto i = (a); i <= (b); ++i)
using namespace std;

const int N = 103, M = 1003;
int n, m, x[N], c[M][3], d[M][3];
mt19937_64 seed(314159);
```

백준 3-SAT 문제

SA로 3-SAT 풀어보기

```cpp
#include <bits/stdc++.h>
#define rep(i,a,b) for (auto i = (a); i <= (b); ++i)
using namespace std;

const int N = 103, M = 1003;
int n, m, x[N], c[M][3], d[M][3];
mt19937_64 seed(314159);

int scoring() {
    int r = 0;
    rep(i,1,m) {
        bool C = false;
        C |= c[i][0] > 0 ? x[c[i][0]] : !x[-c[i][0]];
        C |= c[i][1] > 0 ? x[c[i][1]] : !x[-c[i][1]];
        C |= c[i][2] > 0 ? x[c[i][2]] : !x[-c[i][2]];
        r += C;
    }
    return r;
}

void simulated_annealing(double kt, double d) {
    uniform_int_distribution<int> I(1,n);
    uniform_real_distribution<double> R(0,1);
    double e0 = 0, e1, p;
    rep(_,1,74100) {
        int i = I(seed); x[i] ^= 1;
        e1 = scoring();
        p = exp((e1-e0)/kt);
        if (p < R(seed)) x[i] ^= 1;
        else e0 = e1;
        kt *= d;
        if (e1 == m) {
            cout << "1\n";
            rep(i,1,n) cout << x[i] << ' ';
            exit(0);
        }
    }
}

int main() {
    cin.tie(0)->sync_with_stdio(0);
    cin >> n >> m;
    rep(i,1,m) cin >> d[i][0] >> d[i][1] >> d[i][2];
    rep(_,1,10) {
        rep(i,1,m) rep(j,0,2) c[i][j] = d[i][j];
        fill(x+1,x+n+1,0);
        simulated_annealing(6.5,0.9999);
    }
    cout << 0;
}
```

```cpp
int main() {
    cin.tie(0)->sync_with_stdio(0);
    cin >> n >> m;
    rep(i,1,m) cin >> d[i][0] >> d[i][1] >> d[i][2];
    rep(_,1,10) {
        rep(i,1,m) rep(j,0,2) c[i][j] = d[i][j];
        fill(x+1,x+n+1,0);
        simulated_annealing(6.5);
    }
    cout << 0;
}
```

백준 3-SAT 문제

SA로 3-SAT 풀어보기

```cpp
#include <bits/stdc++.h>
#define rep(i,a,b) for (auto i = (a); i <= (b); ++i)
using namespace std;

const int N = 103, M = 1003;
int n, m, x[N], c[M][3], d[M][3];
mt19937_64 seed(314159);

int scoring() {
    int r = 0;
    rep(i,1,m) {
        bool C = false;
        C |= c[i][0] > 0 ? x[c[i][0]] : !x[-c[i][0]];
        C |= c[i][1] > 0 ? x[c[i][1]] : !x[-c[i][1]];
        C |= c[i][2] > 0 ? x[c[i][2]] : !x[-c[i][2]];
        r += C;
    }
    return r;
}

void simulated_annealing(double kt, double d) {
    uniform_int_distribution<int> I(1,n);
    uniform_real_distribution<double> R(0,1);
    double e0 = 0, e1, p;
    rep(_,1,74100) {
        int i = I(seed); x[i] ^= 1;
        e1 = scoring();
        p = exp((e1-e0)/kt);
        if (p < R(seed)) x[i] ^= 1;
        else e0 = e1;
        kt *= d;
        if (e1 == m) {
            cout << "1\n";
            rep(i,1,n) cout << x[i] << ' ';
            exit(0);
        }
    }
}

int main() {
    cin.tie(0)->sync_with_stdio(0);
    cin >> n >> m;
    rep(i,1,m) cin >> d[i][0] >> d[i][1] >> d[i][2];
    rep(_,1,10) {
        rep(i,1,m) rep(j,0,2) c[i][j] = d[i][j];
        fill(x+1,x+n+1,0);
        simulated_annealing(6.5,0.9999);
    }
    cout << 0;
}
```


```cpp
void simulated_annealing(double kt, double d) {
    uniform_int_distribution<int> I(1,n);
    uniform_real_distribution<double> R(0,1);
    double e0 = 0, e1, p;
    rep(_,1,74100) {
        int i = I(seed); x[i] ^= 1;
        e1 = scoring();
        p = exp((e1-e0)/kt);
        if (p < R(seed)) x[i] ^= 1;
        else e0 = e1;
        kt *= d;
        if (e1 == m) {
            cout << "1\n";
            rep(i,1,n) cout << x[i] << ' ';
            exit(0);
        }
    }
}
```

백준 3-SAT 문제

SA로 3-SAT 풀어보기

```cpp
#include <bits/stdc++.h>
#define rep(i,a,b) for (auto i = (a); i <= (b); ++i)
using namespace std;

const int N = 103, M = 1003;
int n, m, x[N], c[M][3], d[M][3];
mt19937_64 seed(314159);

int scoring() {
    int r = 0;
    rep(i,1,m) {
        bool C = false;
        C |= c[i][0] > 0 ? x[c[i][0]] : !x[-c[i][0]];
        C |= c[i][1] > 0 ? x[c[i][1]] : !x[-c[i][1]];
        C |= c[i][2] > 0 ? x[c[i][2]] : !x[-c[i][2]];
        r += C;
    }
    return r;
}

void simulated_annealing(double kt, double d) {
    uniform_int_distribution<int> I(1,n);
    uniform_real_distribution<double> R(0,1);
    double e0 = 0, e1, p;
    rep(_,1,74100) {
        int i = I(seed); x[i] ^= 1;
        e1 = scoring();
        p = exp((e1-e0)/kt);
        if (p < R(seed)) x[i] ^= 1;
        else e0 = e1;
        kt *= d;
        if (e1 == m) {
            cout << "1\n";
            rep(i,1,n) cout << x[i] << ' ';
            exit(0);
        }
    }
}

int main() {
    cin.tie(0)->sync_with_stdio(0);
    cin >> n >> m;
    rep(i,1,m) cin >> d[i][0] >> d[i][1] >> d[i][2];
    rep(_,1,10) {
        rep(i,1,m) rep(j,0,2) c[i][j] = d[i][j];
        fill(x+1,x+n+1,0);
        simulated_annealing(6.5,0.9999);
    }
    cout << 0;
}
```


```cpp
int scoring() {
    int r = 0;
    rep(i,1,m) {
        bool C = false;
        C |= c[i][0] > 0 ? x[c[i][0]] : !x[-c[i][0]];
        C |= c[i][1] > 0 ? x[c[i][1]] : !x[-c[i][1]];
        C |= c[i][2] > 0 ? x[c[i][2]] : !x[-c[i][2]];
        r += C;
    }
    return r;
}
```

백준 3-SAT 문제

SA로 3-SAT 풀어보기

기하 감률을 했을 때 보다 slow-decrease 감률에다 경곗값을 `5e-4`로 설정했을 때 속도가 3배 넘게 빨라졌지만,
204개의 TC 중 극악의 두 가지 TC는 미처 넘기지 못하는 것으로 확인되었다…

28	`370b704e7a03af269997daab52c525dacd47c7b3`	WA
128	`b37afeb6021a190a8919086e758ec4edf168a97a`	WA

참고

- http://www.aistudy.com/neural/simulated_annealing.htm
- https://en.wikipedia.org/wiki/Simulated_annealing
- https://algospot.com/forum/read/1211/
- https://koosaga.com/3
- https://ryute.tistory.com/35
- https://ryute.tistory.com/36
- https://zzsza.github.io/data/2019/03/26/metaheuristics/
- https://towardsdatascience.com/optimization-techniques-simulated-annealing-d6a4785a1de7
- 이광근, “컴퓨터 과학이 여는 세계”