Анализ на задача "Стена"

Марин Шаламанов, Петър Петров

Март 2018

Нека вземем някой връх от многоъгълника за "начало". Всички точки от стената можем да опишем като, намиращи се на някакво разстояние от началото (движейки се по часовниковата стрелка). Нека точката на разстояние x от началото бележим с $f(x)$. Така например $f(0)$ е самото начало. Ако P е периметърът на многоълника, то винаги $f(x) = f(x + P)$ (защото, ако направим една цяла обиколка на замъка ще се озовем на същото място, откъдето сме тръгнали).

Нека $S(x)$ е използваемостта, за вход намиращ се в точката $f(x)$. Да разгледаме функцията $g(x) = S(x) - S(x + P/2)$. Тази функция е непрекъсната (ако променим много малко x, то $g(x)$ също ще се промени много малко). Също така, ако за някое x имаме $g(x) = 0$, това ще означава, че $S(x) = S(x + P/2)$ и точката x ще е решение на задачата.

Сега да разгледаме $g(0)$ и $g(P/2)$:

$g(0) = S(0) - S(P/2)$

$g(P/2) = S(P/2) - S(P/2 + P/2) = S(P/2) - S(0)$

$\Rightarrow g(0) = -g(P/2)$

Това означава, че в 0 и $P/2$ функцията g има различен знак. Но, тъй като g е непрекъсната, то има точка $x \in [0, P/2]$ за която $g(x) = 0$ и това ще е решение на задачата.

Но как да намерим тази точка? С двоично търсене! Следният код извършва двоично търсене между 0 и $P/2$, за да намери стойността където g е 0.

```
double diff(double pos) {
  double opositePos = pos + per/2.0;
  if (opositePos >= per) opositePos -= per;

  return getUsability(getPoint(pos)) - getUsability(getPoint(
    opositePos));
}

int main() {
  // ...
  double left = 0;
```

```
  double right = per/2;

  if (diff(left) >= 0) swap(left, right);

  // loop invariant diff(left) < 0 & diff(right) >= 0
  while (abs(left - right) > EPS) {
    double mid = (left + right)/2;
    if (diff(mid) < 0) left = mid;
    else right = mid;
  }
}
```

Другият важен въпрос е как да пресмятаме функцията `getUsability` ефективно. Това, което пресмята тази функция е следната сума

$$S(a,b) = \sum_{(x,y)\in houses} (x-a)^2 + (y-b)^2$$

$$= \sum (x^2 - 2xa + a^2 + y^2 - 2yb + b^2) =$$

$$= \sum x^2 + \sum y^2 - 2a(\sum x) - 2b(\sum y) + M(a^2 + b^2)$$

Т.е. е достатъчно предварително да престметнем $\sum x^2$, $\sum y^2$, $\sum x$, $\sum y$ и ще можем за константно време да смятаме `getUsability`.

```
double getUsability(const Point& point) {
  double x = point.x;
  double y = point.y;
  return m*(x*x + y*y) - 2*x*ex - 2*y*ey + ex2 + ey2;
}
```

Последното нещо, което трябва да смятаме бързо е функцията `getPoint(x)`, която ни дава точката, която се намира на разстояние x от "началото"на замъка. Това можем да направим с двоично търсене като превтарително пресметнем за всеки връх от многоъгълника на какво разстояние е от "началото".

Окончателната сложност на алгоритъма е $O(N + M)$.