

Разбор задачи «А. Автобусы»

Автор задачи — К. Батузов, автор разбора — Б. Василевский

Решение на 30 баллов

До начала ремонта дороги интервал между двумя идущими друг за другом автобусами один и тот же для всех автобусов и равен $P = \frac{L}{Mv}$.

Поскольку $v = w$, можно считать, что ремонта дороги не происходило. В этом случае ответ останется равным P .

Решение на 60 баллов

Для прохождения второй группы тестов было необходимо понять, что сразу после начала ремонта интервал между двумя соседними автобусами такой же, каким он будет на протяжении всего дальнейшего пути. Действительно, если нарисовать графики зависимости координаты на маршруте от времени, то можно увидеть следующую картину:

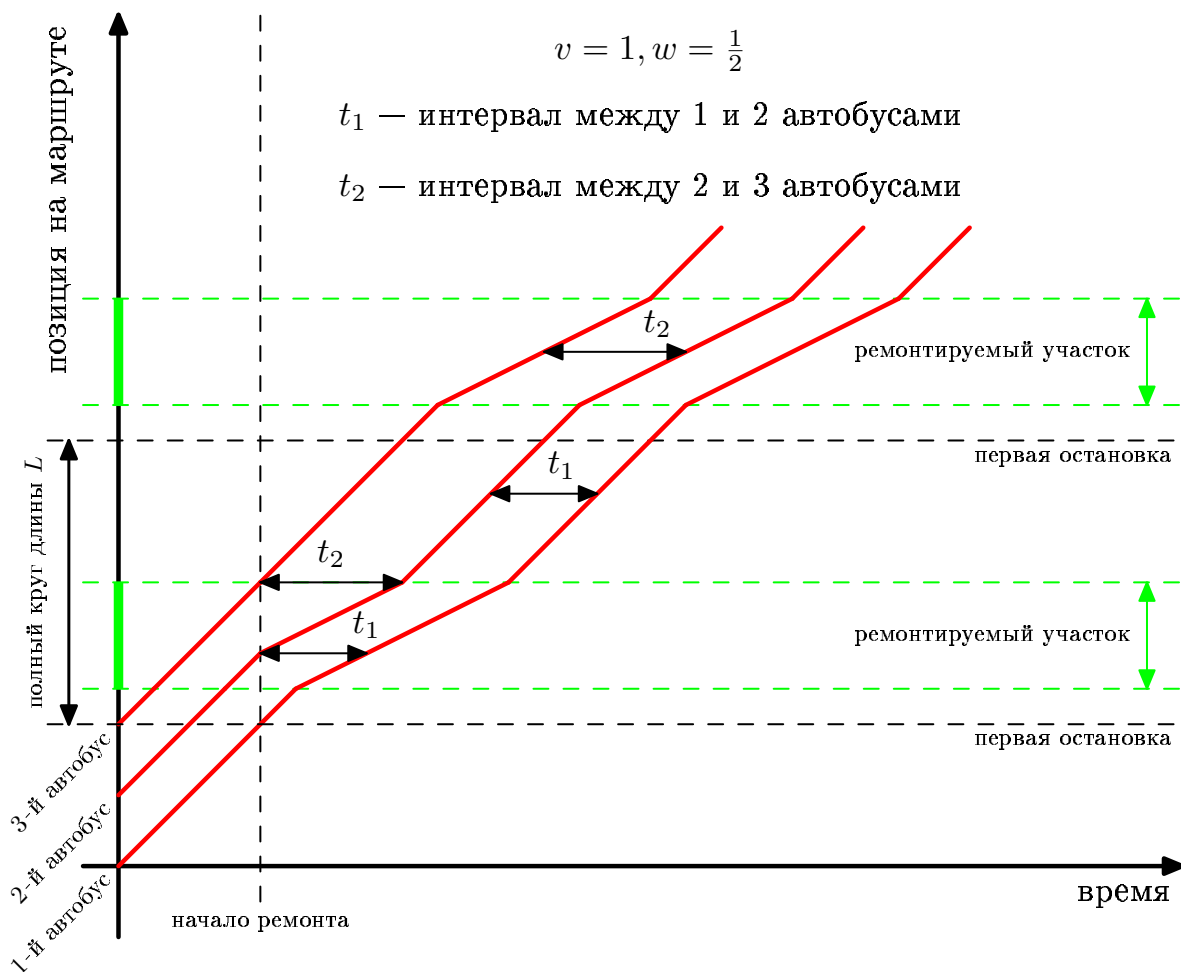


Рис. 1: Интервалы между соседними автобусами можно замерять с момента начала ремонта, ведь в дальнейшем они едут по одному и тому же закону.

Таким образом, для решения задачи нужно узнать максимальный из интервалов между соседними автобусами в тот момент времени, когда начался ремонт.

Исходя из ограничений для второй группы тестов, интервалов всего два. Первый равен в точности времени, за которое первый автобус проедет первую половину окружности (с учетом введенных скоростных ограничений). Вторым определяется аналогично.

Обозначим через d_1 и d_2 длины ремонтируемых участков первой и второй половины пути соответственно, $b = \frac{1}{2}L$. Тогда первый интервал равен в точности $\frac{d_1}{w} + \frac{b-d_1}{v}$, второй — $\frac{d_2}{w} + \frac{b-d_2}{v}$. Следующий отрывок кода является содержательной частью решения, проходящего вторую группу тестов (убедитесь, что d_1 и d_2 вычисляются правильно!):

```

y = y - 1; x = x - 1 // остановки удобно нумеровать с нуля
a = L / N; // переменные a, b, d1, d2 имеют вещественный тип
b = L / M; // деление везде также вещественное
d1 = max(min(a*y, b) - a*x, 0.0);
d2 = max(a*y - max(a*x, b), 0.0);
answer = max(d1 / w + (b - d1) / v, d2 / w + (b - d2) / v);

```

Для того, чтобы набрать 60 баллов, достаточно было в случае $M > 2$ выводить ответ $P = \frac{L}{Mv}$, который работает на первой группе тестов.

Правильное решение

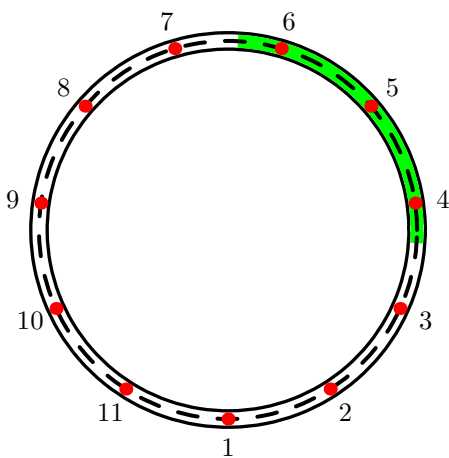


Рис. 2: Случаи 1, 2, 3, 4

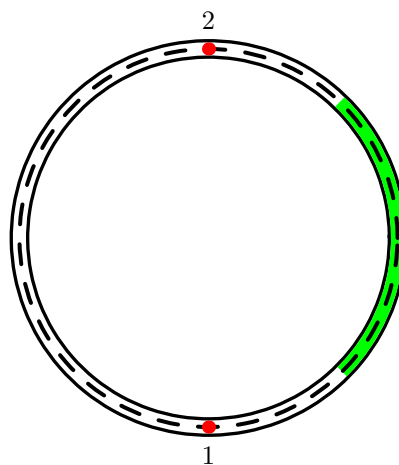


Рис. 3: Случай 5

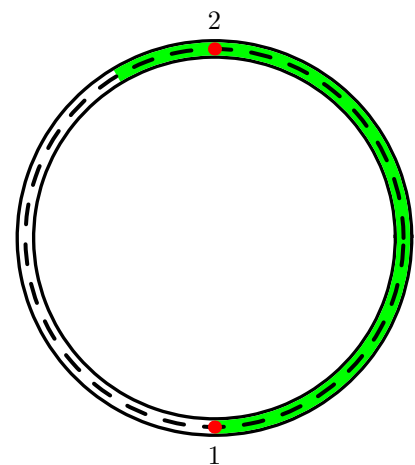


Рис. 4: Случай 4 во втором своем проявлении

Последним логическим шагом в этой задаче является следующее утверждение: всегда существует не более 5 различных интервалов. А именно, это интервалы движения между соседними автобусами, на пути между которыми в момент начала ремонта

- 1) скорость всегда равна v (рис. 2, например, $1 \rightarrow 2$; рис. 3, $2 \rightarrow 1$),
- 2) скорость всегда равна w (рис. 2, например, $4 \rightarrow 5$; рис. 4, $1 \rightarrow 2$),
- 3) скорость сначала равна v , а потом w (рис. 2, $3 \rightarrow 4$),

- 4) скорость сначала равна w , а потом v (рис. 2, $6 \rightarrow 7$; рис. 4, $2 \rightarrow 1$),
- 5) скорость меняется два раза и изначально равна v (рис. 3, $1 \rightarrow 2$),

Случая, когда скорость меняется два раза, но изначально равна w , быть не может, ведь для этого необходимо расположение всех автобусов строго внутри зоны ремонта. Очевидно, что первый автобус может быть либо на границе, либо вне его.

Случаи 3, 4, 5 не могут появляться более одного раза, тогда как 1 и 2, вообще говоря, составляют большинство. Поэтому для каждого из них нужно сначала проверить, что действительно ли он имеет место, и если результат положительный, то вычислить соответствующий интервал.

Несмотря на то, что возможность наличия автобусов на границе ремонтируемого участка практически никак не влияет на реализацию, за этим необходимо аккуратно проследить.

Ниже приведен код решения на си++:

```
#include <algorithm>
#include <cstdio>

using namespace std;

int l, n, m, x, y, v, w;
double answer = 0, d, a, b; // d будет обозначать длину ремонтируемого участка пути

int main()
{
    freopen("a.in", "rt", stdin);
    freopen("a.out", "wt", stdout);
    scanf("%d%d%d%d%d%d", &l, &n, &m, &x, &y, &v, &w);
    x--, y--; // теперь автобусы и остановки нумеруются с нуля

    a = l / (double)n; // расстояние между соседними автобусами
    b = l / (double)m; // расстояние между соседними остановками

    // k1 и k2 - номера первого и последнего автобусов
    //           в участке ремонта в момент его начала
    long long k1 = (long long)m*x;
    k1 = k1 / n + (k1 % n != 0);
    long long k2 = (long long)m*y;
    k2 = k2 / n - (k2 % n == 0);

    if (k2 - k1 + 2 < m && m > 1) // случай 1 имеет место
        answer = max(answer, b/v);

    if (k1 < k2) // случай 2
        answer = max(answer, b/w);

    if (k1 <= k2) // в ремонтируемый участок попал хотя бы один автобус
    {
        if (k2 - k1 + 1 < m) // не все автобусы находятся строго
```

```
{ // внутри ремонтируемого участка
    // случай 3
    d = b*k1 - a*x;
    answer = max(answer, d/w + (b - d)/v);

    // случай 4
    d = a*y - b*k2;
    answer = max(answer, d/w + (b - d)/v);
}
else // случай 4, интервал между k2 и k1
{
    d = (a*y - b*k2) + (b*k1 - a*x);
    answer = max(answer, d/w + (b - d)/v);
}
}
else // случай 5
    answer = max(answer, (b - a*(y - x))/v + a*(y - x)/w);

printf("%.9lf", answer);

return 0;
}
```