



Олимпиада школьников «Гранит науки»
БЛАНК ОЛИМПИАДНОЙ РАБОТЫ

Шифр работы U19-02
(не заполнять)

ВАРИАНТ № 2

Оценка выполнения олимпиадной работы
(заполняется проверяющим)

Первая проверка	№ задания	1	2	3	4	5	6	7	8	9	10	Σ
	Полученный балл	9	4	5	10	19	18	0	0	0	0	68
	Σ баллов прописью	шестьдесят восемь						Подпись проверяющего		[Signature]		
								Подпись проверяющего				
Вторая проверка	№ задания	1	2	3	4	5	6	7	8	9	10	Σ
	Полученный балл	9	7	7	10	19	15					67
	Σ баллов прописью	шестьдесят семь (67)						Подпись проверяющего		[Signature]		
								Подпись проверяющего		[Signature]		



Вариант № 7

~ I

Формулы перетасовали на один столбец назад и на одну строку вниз $\Rightarrow$ там где ссылка не зафиксирована, значение столбца увеличилось на 1, значение строки увеличилось на 1. $\Rightarrow$ $B4 = \text{СЧЁТЕСЛИ}(\$A\$2:\$J\$2; "<0") \Rightarrow$ в $A2:J2$ 4 отрицательных числа; $B5 = B2 * E\$2 / (1\$F2 - H2) = \#\text{ДЕЛ}/0! \Rightarrow F2 - H2 = 0 \Rightarrow F2 = H2$.

$$B6 = \$F\$2 + G\$2 + \$H2 = -8 \Rightarrow 2 \cdot F2 + G2 = -8.$$

$$B8 = \text{ЕСЛИ}(B2+E2; 1; 2) = 2 \Rightarrow B2+E2=0 \Rightarrow B2 = -E2$$

значение если ложь $\Rightarrow$ если 0.

$$B9 = \$A2+B2 = -4 \Rightarrow A2+B2 = -4.$$

$$B10 = E2+D2+I2 = 4$$

$$B11 = E2+D2*I2 = 5$$

$$B12 = \text{ОКРУГВВЕРХ}(\text{СРЗНАЧ}(\$A\$2:I2); 0) - I2 = -3$$

$B7$: т.к. в $A8$ #ССЫЛКА! то дужка дала 1 $\Rightarrow$

$$A2 - F2 + A2 = -6.$$

Вывод: Числа < 0 ; $F2 = H2$; $2F2 + G2 = -8$; $B2 = -E2$; $A2 + B2 = -4$;

$$E2 + D2 + I2 = 4; E2 + D2 * I2 = 5; 2A2 - F2 = -6 \text{ и } \text{ОКРУГВВЕРХ}(\text{СРЗНАЧ}(A2:I2); 0) - I2 = -3.$$

$$D2 + I2 = D2 * I2 - 1.$$

Диаграмма использует строку 1 для легенды, строку 2 для значений $\Rightarrow$ т.к. 3 и 7 не в диаграмме $\Rightarrow C2 = 0$ и $G2 = 0 \Rightarrow$

$$F2 = -4. \Rightarrow H2 = -4; \Rightarrow \text{т.к. сектор 4 в два раза меньше сектора 6} \Rightarrow |D2| = 2 \Rightarrow \text{т.к. } D2 + I2 = D2 * I2 - 1 \text{ если } D2 = -2 \text{ то } I2 = \frac{1}{2} \text{ не подходит} \Rightarrow D2 = 2 \text{ и } I2 = 3. \Rightarrow |E2| = 1 \text{ т.к. сектор меньше чем } \frac{1}{2}; E2 + D2 + I2 = 4 \Rightarrow E2 = -1 \Rightarrow B2 = 1. \Rightarrow A2 = -5. \Rightarrow$$

$$A2:I2 = -5; 1; 0; 2; -1; -4; 0; -4; 3 \Rightarrow \text{ОКРУГВВЕРХ}(\text{СРЗНАЧ}(A2:I2); 0) = \text{ОКРУГВВЕРХ}(-\frac{8}{9}; 0) = 0 \text{ т.к. } -1 = -\frac{9}{9} < -\frac{8}{9} \Rightarrow J2 = 3.$$



Вариант № 4

98

и Ответ: -5; 1; 0; 2; -1; -4; 0; -4; 3; 3² Последнее число найдено неверно

✓3

значение в схеме $(x_2 \wedge x_3) \vee (x_1 \oplus x_2) = (x_2 \wedge x_3) \vee (\bar{x}_1 \wedge x_2) \vee (x_1 \wedge \bar{x}_2)$
 $x_1 \vee x_2 \wedge x_3 \oplus x_2 \wedge x_3$

⇒ Таблица:

x_1	x_2	x_3	знач.
0	0	0	0
0	0	1	0
0	1	0	1
0	1	1	1
1	0	0	1
1	0	1	1
1	1	0	0
1	1	1	0

78

⇒ набори (0; 0; 0), (0; 0; 1), (1; 1; 0).

Для них выражение:

0; 0; 0 : $(0 \rightarrow 0) \Leftrightarrow (1 \oplus 0)$ $1 \oplus 1$ истинно 1. + 0, 1, 0

0; 0; 1 : $(0 \rightarrow 0) \Leftrightarrow (0 \oplus 0)$ $1 \Leftrightarrow 0$ ложь 0. + 0, 1, 1

- 1; 1; 0 : $(1 \rightarrow 1) \Leftrightarrow (1 \oplus 1)$ $1 \Leftrightarrow 0$ ложь 0. + 1, 1, 1

Ответ: 100 / 10011

✓4

рис 1.

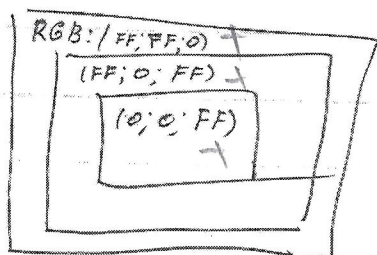
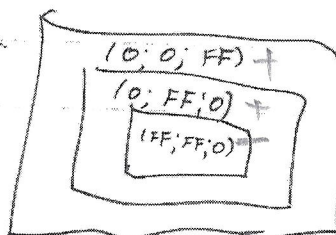


рис 2.



8+4 = (100)

Алгоритм обработки - универсал: для каждого канала яркость с x на FF-x.

1. Нет теоретического обоснования алгоритма
2. Значения яркости в RGB формате (255; 255; 0) не представлено



Вариант № 4

N5

Язык пython.

+

$N, R, D = \text{map}(\text{int}, \text{input}().\text{split}())$

$G = N - R$

$R1 = 0$

$G1 = 0$

for i in range(D):

~~R1 = R1 + G~~

(Tab) $G1 = G1 + R1 \cdot G$

(Tab) $R1 = R1 \cdot R$

(Tab) $R1 = R1 \% 70000000007$

(Tab) $G1 = G1 \% 70000000007$

print(R1, G1).

Для исходных данных: 4096 6825

N2

Для двух кучных куч: как бы не сходил первый, второй может одна из куч будет с нечётным количеством камней, второй может поменять чётность в одной или в обеих кучах $\Rightarrow$ сделать так, чтобы после его хода в обеих кучах было чётное число камней $\Rightarrow (0;0)$ может получиться после хода второго. Пусть n -чётное кол-во камней, r -чётное $\Rightarrow$ ход первого $(r; r) \xrightarrow{(1;1)} (n; n) \xrightarrow{(1;1)} (n; r) \xrightarrow{(0;1)} (r; n) \Rightarrow$ ход второго:

$(n; n) \xrightarrow{(1;1)} (r; r)$

$(n; r) \xrightarrow{(1;0)} (r; r)$

$(r; n) \xrightarrow{(0;1)} (r; r)$

ход второго сделан т.к. $n \neq 0$.

Нет графового полиномиального алгоритма, с использованием сжатых графов и указателем пропускной способности.



Вариант № 7

№2 (продолж.)

Алгоритм для любого числа камней.

твой ход

составь кучу $(1; 1)$; $(1; 1)$; $(1; 1)$; $(1; 1)$; $(0; 0)$

↓
двигай-то
двигай-то
проиграешь
отдай ход

$(1; 1)$ →
отдай
ход

$(1; 1)$ →
отдай
ход

$(0; 1)$ →
отдай
ход

мы проиграли.

В решении $(x; y)$ - составь кучу; $(1; 1)$ - забрать камень из кучи 1; $(0; 1)$ - забрать камень из кучи 2.
 $(1; 1)$ - забрать по камню из обеих.

№6. +

Для исходных значений 2. Решение на Python.

```
n, m = map(int, input().split())
```

```
a = []
```

```
for i in range(n):
```

```
    (Tab) q = input()
```

```
    (Tab) a.append(q)
```

```
    (Tab) for j in q:
```

```
        (Tabx2) if j == ".":
```

```
            (Tabx3) a[-1].append(0)
```

```
        (Tabx2) else:
```

```
            (Tabx3) a[-1].append(1)
```

~~≠ a - наша очередь, если же пришла очередь к противнику~~

```
dm = [0] * n
```

```
for i in range(n):
```

```
    (Tab) dm.append([0] * (m+1))
```



Вариант № 7

~ 6 (продолжение)

```
for i in range(n):
    (Tab) for j in range(m):
        (Tabx2) if a[i][j] != 1:
            (Tabx3) cn = 0
            (Tabx3) re = 1
            (Tabx3) for while re == 1:
                (Tabx4) cn += 1
                (Tabx4) for it in range(cn, cn):
                    (Tabx5) for jt in range(cn, cn):
                        (Tabx6) j+it < m and
                        if a[i+it][j+jt] == 1:
                            (Tabx7) re = 0
                            (Tabx7) break
            (Tabx3) a[i][j] = cn - 1
# ат-возвращаем индекс размер подомы с верхним левым углом здесь.
ans = n-1
tans = 0
while tans == 0:
    (Tabx1) ans -= 1
    tans = f(at, ans)
print(ans)
```

дальше идёт описание f, в программе оно должно быть в начале, f-функция, которая определяет, можно ли такой ход проехать.



Вариант № 7

~ 6 (продолжение ~)

def f(a1, ans, n):

(Tabx1) nor = []

(Tabx1) for i in range(n):

(Tabx2) if a1[i] > ans:

(Tabx3) nor.append(i)

(Tabx1) for ik in range(1, len(a1[0])):

(Tabx2) norm = []

(Tabx2) for i in nor:

(Tabx3) if a1[i][ik] > ans:

(Tabx4) norm.append(i)

(Tabx2) for i in range(len(norm)):

(Tabx3) if a1[norm[i]-1][ik] > ans and norm[i]-1 not in norm:

(Tabx4) norm.append(norm[i]-1) # сюда 0 ans ans

(Tabx3) if a1[norm[i]+1][ik] > ans and norm[i]+1 not in norm:

(Tabx4) norm.append(norm[i]+1)

(Tabx2) nor = norm. # nor можем вернуть на этом ходу.

(Tabx1) if len(nor) > 0: # мы можем вернуть в конце.

(Tabx2) return 1

(Tabx1) return 0

конец.

f - просто модифицированный валидный алгоритм.