

## §2 Второй отборочный этап

Второй отборочный этап проводится в командном формате в сети интернет, работы оцениваются автоматически средствами системы онлайн-тестирования. Продолжительность второго отборочного этапа — 2 недели. Задачи носят междисциплинарный характер и в более простой форме воссоздают инженерную задачу заключительного этапа. Решение каждой задачи дает определенное количество баллов. Баллы зачисляются в полном объеме за правильное решение задачи. В данном этапе можно получить суммарно от 0 до 8 баллов.

### 2.1. Задачи по энергетике

#### Задача 2.1.1 (1 балл)

В таблице приведены зависимости средних значений генерации и потребления электроэнергии от времени в течение суток в локальной энергосистеме поселка в Крыму в МВт/час. Определите, какая минимальная ёмкость накопителей электроэнергии необходима для того, чтобы энергосистема полностью обеспечивалась собственной генерацией?

В ответ запишите следующие значения:

begin =

Час, с которого начинается период профицита, в двадцатичетырехчасовом формате, без минут, например, 13.

end =

Час, с которого начинает период дефицита энергии в двадцатичетырехчасовом формате, без минут, например, 21.

capacity =

Необходимую емкость накопителей электроэнергии в МВт/час, число необходимо дать с точностью до второго знака, для отделения дробной части используется точка.

#### Пример входных данных:

Время	Расход	Производство
00:00–01:00	1.05	0.07
01:00–02:00	0.57	0.04
02:00–03:00	0.35	0.07
03:00–04:00	0.47	0.20
04:00–05:00	0.63	0.32
05:00–06:00	0.91	0.41
06:00–07:00	1.26	0.50
07:00–08:00	1.74	0.76

08:00-09:00	2.32	1.15
09:00-10:00	2.68	1.75
10:00-11:00	2.70	2.43
11:00-12:00	2.31	3.03
12:00-13:00	1.82	3.35
13:00-14:00	1.48	3.54
14:00-15:00	1.14	3.74
15:00-16:00	0.96	3.79
16:00-17:00	0.80	3.44
17:00-18:00	0.95	3.01
18:00-19:00	1.40	2.26
19:00-20:00	1.91	1.72
20:00-21:00	2.42	1.01
21:00-22:00	2.52	0.66
22:00-23:00	2.32	0.34
23:00-24:00	1.76	0.17

### СПОСОБ ОЦЕНКИ РАБОТЫ:

Для генерации уникального условия и проверки результата используется следующий код на языке Python. Функция generate возвращает варианты набора исходных данных:

```

cons_base = [0.2, 0.3, 0.4, 0.55, 0.7, 1.1, 1.6, 2.1, 2.3, 2.3, 2.0, 1.4, 1.1,
0.9, 0.7, 0.6, 0.7, 1.2, 1.9, 2.1, 2.1, 1.9, 1.5, 0.9 ]
gen_base = [0.0, 0.0, 0.0, 0.2, 0.25, 0.3, 0.5, 0.9, 1.4, 2.2, 2.8, 3.0, 3.0,
3.0, 3.0, 2.7, 2.3, 1.9, 1.1, 0.6, 0.4, 0.2, 0.1, 0.0]

def avg(arr, n):
    ret = []
    for idx, i in enumerate(arr):
        ret.append(sum([arr[idx - i] for i in range(n)])/n)
    return ret

def rad():
    return abs(rd.random() * 0.1)

def gauss(x, mu, s):
    return 1./(s * (2.*math.pi)**1/2.) * math.e ** ( - (x - mu)**2./ (2. * s))

def letgen(cons_base, gen_base):
    cons = []
    gen = []
    while (sum(gen) <= sum(cons)):
        cons = [round((cons_base[i] + (gauss(i, rd.randint(7,11), 3) + gauss(i,
rd.randint(18,22), 3)) * 5 + rad() * 2),2) for i in range(len(cons_base))]
        k = 1.1
        gen = [round(((gen_base[i] + gauss(i, rd.randint(13,17), 3) * 5 +
rad()*2) * k), 2) for i in range(len(cons_base))]

        cons = avg(cons, 3)
        gen = avg(gen, 3)

    return [cons, gen]

```

```

def defc(gen, cons):
    idxs = []
    defs = []
    for idx, val in enumerate(gen):
        if val < cons[idx]:
            idxs.append(idx)
            defs.append(val - cons[idx])
    return [idxs, defs]

def prof(gen, cons):
    idxs = []
    pros = []
    for idx, val in enumerate(gen):
        if val > cons[idx]:
            idxs.append(idx)
            pros.append(val - cons[idx])
    return [idxs, pros]

def generate():
    num_tests = 10
    tests = []
    string = ""
    [cons, gen] = letgen(cons_base, gen_base)
    for idx, val in enumerate(cons):
        string += "{0:02d}:00-{3:02d}:00\t{1:.2f}\t\t{2:.2f}\n".format(idx, val,
gen[idx], idx+1)
    string = "Время\t\tРасход\t\tПроизводство\n" + string

    tests.append(string)

    return tests

def check(reply, clue):
    return str(reply) == str(clue)

```

### **РЕШЕНИЕ:**

Задача имеет следующий алгоритм решения:

1. При помощи таблицы определить суммарные потребление и генерацию электроэнергии. Вычислить избыток суточной мощности в системе.
2. Найти временные интервалы, в которых система испытывает недостаток электроэнергии. Вычислить недостаток энергии в течение каждого из этих интервалов.
3. Найти временные интервалы, в которых в системе существует профицит энергии. Вычислить избыток энергии во время этих интервалов.
4. Для каждого энергодефицитного интервала определить, достаточно ли избытка энергии предыдущего профицитного интервала для покрытия недостатка энергии во время следующего дефицитного. Если это не так, то часть запаса энергии в накопителях должна остаться с предыдущего дефицитного интервала.
5. Минимальная необходимая мощность накопителей — та, которая позволит покрыть дефицит энергии во время всех дефицитных интервалов.

### Задача 2.1.2 (2 балла)

Для описанной в задаче 2.1.1. локальной энергосистемы определите минимально необходимую для автономного существования энергосистемы величину мощности накопителей электроэнергии, если известно, что за каждый час эти накопители разряжаются на 10%.

Данные приведены в таблице, аналогичной таблице из задачи 2.1.1.

Полученное значение запишите с точностью до двух знаков после запятой в строку capacity =

#### Пример входных данных:

Время	Расход	Производство
00:00-01:00	1.07	0.16
01:00-02:00	0.58	0.08
02:00-03:00	0.43	0.10
03:00-04:00	0.58	0.18
04:00-05:00	0.69	0.27
05:00-06:00	0.93	0.36
06:00-07:00	1.26	0.49
07:00-08:00	1.90	0.75
08:00-09:00	2.44	1.13
09:00-10:00	2.82	1.75
10:00-11:00	2.62	2.49
11:00-12:00	2.25	3.12
12:00-13:00	1.72	3.41
13:00-14:00	1.35	3.43
14:00-15:00	0.99	3.48
15:00-16:00	0.83	3.43
16:00-17:00	0.74	3.41
17:00-18:00	0.97	3.16
18:00-19:00	1.55	2.65
19:00-20:00	2.05	1.82
20:00-21:00	2.48	1.07
21:00-22:00	2.37	0.58
22:00-23:00	2.29	0.38
23:00-24:00	1.72	0.21

#### СПОСОБ ОЦЕНКИ РАБОТЫ:

Для генерации уникального условия и проверки результата используется следующий код на языке Python. Функция generate возвращает варианты набора исходных данных:

```
cons_base = [0.2, 0.3, 0.4, 0.55, 0.7, 1.1, 1.6, 2.1, 2.3, 2.3, 2.0, 1.4, 1.1,
```

```

0.9, 0.7, 0.6, 0.7, 1.2, 1.9, 2.1, 2.1, 1.9, 1.5, 0.9 ]
gen_base = [0.0, 0.0, 0.0, 0.2, 0.25, 0.3, 0.5, 0.9, 1.4, 2.2, 2.8, 3.0, 3.0,
3.0, 3.0, 2.7, 2.3, 1.9, 1.1, 0.6, 0.4, 0.2, 0.1, 0.0]

def avg(arr, n):
    ret = []
    for idx, i in enumerate(arr):
        ret.append(sum([arr[idx - i] for i in range(n)])/n)
    return ret

def rad():
    return abs(rd.random() * 0.1)

def gauss(x, mu, s):
    return 1./(s * (2.*math.pi)**1/2.) * math.e ** ( - (x - mu)**2./ (2. * s))

def letgen(cons_base, gen_base):
    cons = []
    gen = []
    while (sum(gen) <= sum(cons)):
        cons = [round((cons_base[i] + (gauss(i, rd.randint(7,11), 3) + gauss(i,
rd.randint(18,22), 3)) * 5 + rad() * 2),2) for i in range(len(cons_base))]
        k = 1.1
        gen = [round(((gen_base[i] + gauss(i, rd.randint(13,17), 3) * 5 +
rad()*2) * k), 2) for i in range(len(cons_base))]

        cons = avg(cons, 3)
        gen = avg(gen, 3)

    return [cons, gen]

def defc(gen, cons):
    idxs = []
    defs = []
    for idx, val in enumerate(gen):
        if val < cons[idx]:
            idxs.append(idx)
            defs.append(val - cons[idx])
    return [idxs, defs]

def prof(gen, cons):
    idxs = []
    pros = []
    for idx, val in enumerate(gen):
        if val > cons[idx]:
            idxs.append(idx)
            pros.append(val - cons[idx])
    return [idxs, pros]

def generate():
    num_tests = 10
    tests = []
    string = ""
    [cons, gen] = letgen(cons_base, gen_base)
    for idx, val in enumerate(cons):
        string += "{0:02d}:00-{3:02d}:00\t{1:.2f}\t\t{2:.2f}\n".format(idx, val,
gen[idx], idx+1)
        string = "Время\t\tРасход\t\tПроизводство\n" + string

    tests.append(string)

```

return tests

### РЕШЕНИЕ:

Задача имеет следующий алгоритм решения:

1. Вычислить энергию, которую может накопить система за каждый профицитный интервал. Поскольку данные дискретны, возможно считать, что накопитель теряет 10% заряда сразу после дозарядки, либо по прошествии дополнительного часа.

2. Вычислить минимально необходимый запас энергии в накопителях, с которым система должна подойти к энергодефицитному интервалу, чтобы полностью его покрыть. Сделать нужно для каждого интервала. Аналогично, накопитель разряжается после прошествия периода (1 часа) разрядки, либо до.

### Задача 2.1.3 (2 балла)

Спроектируйте систему лопастных ветрогенераторов (ветроэнергетических установок – ВЭУ) с горизонтальной осью вращения для энергоснабжения прибрежного военного городка, потребляющего заданную в условиях мощность. В ответе укажите минимально необходимое число ветрогенераторов и высоту расположения оси ротора.

Считайте все ВЭУ одинаковыми по конструкции и характеристикам, они вырабатывают одинаковую мощность при равных условиях работы. Максимально допустимая высота расположения оси ротора — 50 м. Диаметр лопастей ротора — 6 м. Коэффициент мощности (доля получаемой ветрогенератором энергии воздушного потока) — 0,45. КПД трансмиссии — 0,9. КПД электрогенератора — 0,96.

Скорость ветра зависит от высоты. Эта зависимость соответствует формуле

$$v = v_0 \left( \frac{h}{10} \right)^{\frac{1}{3}},$$

где  $h$  — высота над уровнем поверхности,  $v_0$  — скорость ветра на высоте 10 м в м/с. В условиях этой задачи можно считать что скорость ветра постоянна. Плотность воздуха считайте равной 1,226 кг/м<sup>3</sup>.

Ниже приведены требуемая мощность в Ваттах и скорость ветра в метрах в секунду на высоте 10 м. над землей. В качестве ответа приведите необходимое количество ветряков и их высоту. Запишите ответы в строки `quantity` и `height` соответственно.

### Пример входных данных:

99000.0 3.35

### РЕШЕНИЕ:

Ветроэнергетическая установка преобразует мощность ветрового потока в

электрическую. Энергия проходит путь: ветер — ротор — трансмиссия — генератор. На каждом участке часть энергии теряется в соответствии с его КПД. Из мощности ветрового потока в результате преобразования получается электрическая мощность.

Мощность ветрового потока можно получить из его кинетической энергии:

$$E_k = \frac{1}{2} m v^2 \frac{1}{t}, \text{ где } t \text{ — некоторый временной интервал.}$$

Масса воздуха, проходящая через сечение ротора за интервал времени  $t$ , составляет

$$m = \rho v \frac{\pi d^2}{4} \cdot t.$$

Таким образом, мощность ветра, проходящего через ротор, можно выразить по формуле  $P_v = \rho v^3 \frac{\pi d^2}{8}$ .

После всех преобразований энергии мощность генератора составит

$$P_w = \rho v^3 \frac{\pi d^2}{8} \cdot \eta_{\text{вз}} \cdot \eta_{\text{мп}} \cdot \eta_{\text{эз}}.$$

Теперь нужно определить скорость ветра на той высоте, на которой находится ротор. Из заданной функции скорость ветра всегда возрастает с высотой, поэтому ротор разумно расположить на максимальной доступной высоте — 50 метров. Скорость ветра на высоте 50 метров в  $\sqrt[3]{5}$  раз больше, чем на высоте 10 метров, и составит примерно 5,1 м/с. Отсюда мощность ветрогенератора составит 893 Вт. Для обеспечения нужд посёлка понадобится 112 таких генераторов.

### Задача 2.1.4 (3 балла)

Вам нужно выбрать оптимальные параметры для некоторой (несуществующей) электростанции. Её КПД можно весьма точно оценить по формуле:

$$(1.16 + (a - 1.6x)^{1.9} + b(38.9 + y - (x + 2.2)^{2.6})^2)^{-0.3}$$

Параметрами  $x$  и  $y$  вы можете управлять. Значения  $x$  ограничены от 1 до 3, значения  $y$  — от 2 до 5.

Найдите такие значения параметров  $x$  и  $y$ , при которых КПД электростанции будет оптимальным при следующих параметрах  $a$  и  $b$ . Дайте ответ с точностью до третьего знака после запятой.

**Пример входных данных:**

$$a = 4.03 \quad b = 74.64$$

## **РЕШЕНИЕ:**

КПД условной электростанции желательно должен быть максимально большим, таким образом, при заданных значениях параметров  $a$  и  $b$  задача сводится к отысканию максимума функции КПД.

Поскольку при решении задачи можно дать решение с точностью до третьего знака, задача может быть решена численно. Например, в MS Excel или с помощью написания простой программы эта задача может быть решена напрямую построением таблицы значений функции КПД на сетке значений дискретных переменных с дальнейшим нахождением наибольшего из значений функции.

Таким образом можно найти, что КПД максимально при значениях  $x = 2,082$ ;  $y = 4,981$ .

## **2.2. Критерии определения призеров и победителей**

Количество баллов, набранных при решении всех задач суммируется. Призерам второго отборочного этапа было необходимо набрать 1 балл. Победители второго отборочного этапа должны были набрать 8 баллов.