Assignment 2.1: Changes (3 points)

An array of n number of integers must be modified so that no two consecutive integers are equal. The new value can be chosen arbitrarily. What is the minimum number of required changes?

For Example array $[1, 1, 2, 2, 2]$ requires 2 changes. Changed array can be e.g. $[1, 3, 2, 3, 2]$.

Create the following function in Python:

- `changes(A: list)`: returns the minimum number of required changes

Limits:

- $1 \leq n \leq 10^5$
- each integer is between 1^3

Targets:

1. produces correct solution: **+2 points**
2. has time complexity of $\Theta(n)$: **+1 points**

A code template with an example program:

```
def changes(A):
    # TODO


if __name__ == "__main__":
    print(changes([1, 1, 2, 2, 2]))
    print(changes([1, 2, 3, 4, 5]))
    print(changes([1, 1, 1, 1, 1]))
```

Output

```
$ python changes.py
2
0
2
```

Assignment 2.2: Bit Pairs (3 points)

We are given a bit string which each character is either 0 or 1. Count the sum of each distance of bit pairs where both bits are 1.

For example a bit string 100101 has following distances

- **1**00**1**01 (3)
- **1**0010**1** (5)
- 100**1**0**1** (2)

Therefore the sum of distances is $3 + 5 + 2 = 10$.

Create the following function in Python:

- `pairs(s: str)`: returns the sum of distances

Limits: the maximum length of the bit string is 10^5

Targets:

1. produces correct solution: **+2 points**
2. has time complexity of $\Theta(n)$: **+1 points**

A code template with an example program:

```
def pairs(s):
    # TODO


if __name__ == "__main__":
    print(pairs("100101"))
    print(pairs("101"))
    print(pairs("100100111001"))
```

Output

```
$ python bitpairs.py
10
2
71
```

Submit your solution solution in CodeGrade as `bitpairs.py`.

Assignment 2.3: Split Lists (3 points)

An array of n number of integers must be split in two sub arrays so that every integer of left sub array are smaller than every integer of right sub array. In how many points the array can be split in half?

For example array $[2, 1, 2, 5, 7, 6, 9]$ can be split in 3 ways:

1. $[2, 1, 2]$ and $[5, 7, 6, 9]$
2. $[2, 1, 2, 5]$ and $[7, 6, 9]$
3. $[2, 1, 2, 5, 7, 6]$ and $[9]$

Create following function(s) in Python:

- `split(A: list)`: returns the number of possible splits

Limits:

Targets:

1. produces correct solution: **+2 points**
2. has time complexity of $\Theta(n)$: **+1 points**

A code template with an example program:

```
def split(T):
    # TODO



if __name__ == "__main__":
    print(split([1, 2, 3, 4, 5]))
    print(split([5, 4, 3, 2, 1]))
    print(split([2, 1, 2, 5, 7, 6, 9]))
    print(split([1, 2, 3, 1]))
```

Output

```
$ python bitpairs.py
4
0
3
0
```

Submit your solution solution in CodeGrade as `split.py`.

Last modified: Wednesday, 3 July 2024, 1:23 AM