

Puzzle

Friday, June 9, 2017 5:40 PM

Suppose you're editing a text file from a blank text editor. You have a keyboard with 4 keys: select all, copy, paste, and type 1 character.

Given n allowed keystrokes, find the sequence of keystrokes which maximizes the total number of characters in the end. First solve the simpler problem of finding how MANY characters you can create with n keystrokes. Backtrack the solution later to figure out the sequence of keystrokes.

Define $c[n]$ to be the optimal number of CHARACTERS given n allowed keystrokes.

| Action | Letter |
|-----------------|--------|
| Select all | S |
| Copy selection | C |
| Paste clipboard | P |
| Type 1 letter | T |

Base cases:

$c[0] = 0$
 $c[1] = 1, T$
 $c[2] = 2, TT$
 $c[3] = 3, TTT$
 $c[4] = 4, TTTT$

At some step n , there are 2 things I can do. I could type 1 character, or I could select all, copy, and paste a certain number of times. However, there must be a limit on the number of consecutive pastes we can do before it becomes more efficient to do another select all and copy to increase the number of letters in our clipboard. However, we know that there must be at least 1 paste. If we selected all and copied, it would be inefficient to type 1 character, since the clipboard has more than 1 character (assuming non base case).

How many times can we paste consecutively before it becomes inefficient?

| | |
|--|---|
| <p>S C P P P P P</p> <p>Final amount on clipboard: 1</p> | <p>Conclusion: it makes no sense to paste more than 4 times in a row, since we can just achieve the final result in a different way, but with more letters on our clipboard</p> |
| <p>S C P S C P P</p> <p>2</p> | |
| <p>S C P P S C P</p> <p>3</p> | |

Therefore, at a particular step, I have 5 possible move sequences:

| Type 1 character | $c[n] = c[n - 1] + 1$ | | | | | | | | | | | | | | | | | | |
|---------------------------------|--|--------------|--------------|--------------|--------------|--------------|----------|--------|-------|-------|-------|-------|-------|----|-----|------|-----|------|-------|
| Select all, copy, paste 1 time | <table><tr><th>Step $n - 2$</th><th>Step $n - 1$</th><th>Step n</th></tr><tr><td>Select</td><td>Copy</td><td>Paste</td></tr><tr><td>X</td><td>X</td><td>XX</td></tr></table> <p>(assuming at the end of step $n - 3$, we have $c[n - 3] = X$ letters available) $c[n] = c[n - 3] + c[n - 3] = 2c[n - 3]$</p> | Step $n - 2$ | Step $n - 1$ | Step n | Select | Copy | Paste | X | X | XX | | | | | | | | | |
| Step $n - 2$ | Step $n - 1$ | Step n | | | | | | | | | | | | | | | | | |
| Select | Copy | Paste | | | | | | | | | | | | | | | | | |
| X | X | XX | | | | | | | | | | | | | | | | | |
| Select all, copy, paste 2 times | <table><tr><th>Step $n - 3$</th><th>Step $n - 2$</th><th>Step $n - 1$</th><th>Step n</th></tr><tr><td>Select</td><td>Copy</td><td>Paste</td><td>Paste</td></tr><tr><td>X</td><td>X</td><td>XX</td><td>XXX</td></tr></table> <p>$c[n] = c[n - 4] + c[n - 4] + c[n - 4] = 3c[n - 4]$</p> | Step $n - 3$ | Step $n - 2$ | Step $n - 1$ | Step n | Select | Copy | Paste | Paste | X | X | XX | XXX | | | | | | |
| Step $n - 3$ | Step $n - 2$ | Step $n - 1$ | Step n | | | | | | | | | | | | | | | | |
| Select | Copy | Paste | Paste | | | | | | | | | | | | | | | | |
| X | X | XX | XXX | | | | | | | | | | | | | | | | |
| Select all, copy, paste 3 times | <table><tr><th>Step $n - 4$</th><th>Step $n - 3$</th><th>Step $n - 2$</th><th>Step $n - 1$</th><th>Step n</th></tr><tr><td>Select</td><td>Copy</td><td>Paste</td><td>Paste</td><td>Paste</td></tr><tr><td>X</td><td>X</td><td>XX</td><td>XXX</td><td>XXXX</td></tr></table> <p>$c[n] = c[n - 5] + c[n - 5] + c[n - 5] + c[n - 5] = 4c[n - 5]$</p> | Step $n - 4$ | Step $n - 3$ | Step $n - 2$ | Step $n - 1$ | Step n | Select | Copy | Paste | Paste | Paste | X | X | XX | XXX | XXXX | | | |
| Step $n - 4$ | Step $n - 3$ | Step $n - 2$ | Step $n - 1$ | Step n | | | | | | | | | | | | | | | |
| Select | Copy | Paste | Paste | Paste | | | | | | | | | | | | | | | |
| X | X | XX | XXX | XXXX | | | | | | | | | | | | | | | |
| Select all, copy, paste 4 times | <table><tr><th>Step $n - 5$</th><th>Step $n - 4$</th><th>Step $n - 3$</th><th>Step $n - 2$</th><th>Step $n - 1$</th><th>Step n</th></tr><tr><td>Select</td><td>Copy</td><td>Paste</td><td>Paste</td><td>Paste</td><td>Paste</td></tr><tr><td>X</td><td>X</td><td>XX</td><td>XXX</td><td>XXXX</td><td>XXXXX</td></tr></table> <p>$c[n] = c[n - 6] + c[n - 6] + c[n - 6] + c[n - 6] + c[n - 6] = 5c[n - 6]$</p> | Step $n - 5$ | Step $n - 4$ | Step $n - 3$ | Step $n - 2$ | Step $n - 1$ | Step n | Select | Copy | Paste | Paste | Paste | Paste | X | X | XX | XXX | XXXX | XXXXX |
| Step $n - 5$ | Step $n - 4$ | Step $n - 3$ | Step $n - 2$ | Step $n - 1$ | Step n | | | | | | | | | | | | | | |
| Select | Copy | Paste | Paste | Paste | Paste | | | | | | | | | | | | | | |
| X | X | XX | XXX | XXXX | XXXXX | | | | | | | | | | | | | | |

$$c[n] = \max\left(\max_{2 \leq k \leq 5} \{k * c[n - k - 1]\}, c[n - 1] + 1\right)$$

Recursive solution

| | |
|--|---|
| <pre>def f(n): if n<=4: return n x = f(n-1)+1 for i in range(2, 5): x = max(x, i*f(n-i-1)) return x</pre> | Value in $c[n]$ depends on the immediate 6 values behind it |
|--|---|

Dynamic programming solution

| | |
|---|---|
| <pre>def f(n): c = ([0, 1, 2, 3, 4] + [0]*(n+1-5)) for k in range(5, n+1): x = c[k-1]+1 for i in range(2, 6): x = max(x, i*c[max(k-i-1, 0)]) c[k] = x return c[n]</pre> | <p>$c = [0, 1, 2, 3, 4, 0\dots]$</p> <p>Loop k from [5, n]</p> <p>Loop i from [2, 5]</p> <p>Apply the recurrence</p> |
|---|---|

Constructing the solution

```
def print_if(s, b):
    if b:
        print(s)

def f(n):
    c = ([0, 1, 2, 3, 4] + [0]*(n+1-5))
    for _ in range(min(n, 4)):
        print("Type 1 character")

    for k in range(5, n+1):
        x = c[k-1]+1
        val_if_type = x
        case = -1
        y = x
        for i in range(2, 6):
            x = max(x, i*c[max(k-i-1, 0)])
            if x > y:
                y = x
                case = i
        for r in range(case):
            print_if("Select all", k==n)
            print_if("Copy selection", k==n)
            for _ in range(case-1):
                print_if("Paste", k==n)
        c[k] = x
        if val_if_type == x:
            print_if("Type 1 character", k==n)
    return c[n]
```