Cookie Clicker

Kai Xiao
kaix@mit.edu

November 26, 2016

Abstract

Cookie Clicker is a popular online incremental game. The goal of the game is to bake as many cookies as possible by clicking on cookies or by using cookies as currency to buy items and upgrades that increase cookie generation rate. In this paper, we analyze strategies for playing Cookie Clicker optimally. While simple to state, the game gives rise to interesting analysis involving ideas from NP-hardness, approximation algorithms, and dynamic programming. You can try playing Cookie Clicker yourself here: http://orteil.dashnet.org/cookieclicker/.

Contents

1 Introduction **2**

2 Basic Example: 1 Item Cookie Clicker **2**
- 2.1 Model . . . 2
- 2.2 Useful tools . . . 2
- 2.3 Solution . . . 4

3 Positive Results **5**
- 3.1 Fixed Cost Cookie Clicker for 2 Items . . . 5
- 3.2 Fixed Cost Cookie Clicker for n Items . . . 8
  - 3.2.1 Dynamic Programming Solution . . . 9
  - 3.2.2 Local Optimizations . . . 9
- 3.3 Increasing Cost Cookie Clicker (n-item) . . . 9
  - 3.3.1 Natural Greedy Solution . . . 9
  - 3.3.2 Frozen Cookie Greedy Solution . . . 9
  - 3.3.3 Approximation Ratio . . . 9

4 Negative Results **11**
- 4.1 General Cookie Clicker . . . 11
  - 4.1.1 Dynamic Programming Solution . . . 12
  - 4.1.2 NP-hardness of the General Cookie Clicker problem . . . 13
- 4.2 Discrete Cookie Clicker . . . 15

5 Code **16**

6 Conclusion **16**

1 Introduction

In Cookie Clicker, your goal is to bake as many cookies as possible. You can click on a big cookie icon to bake a cookie, and you can also obtain items that automatically bakes cookies for you over time. You can use the cookies you have baked to purchase items, such as grandmas and factories, that generate a certain number of cookies per second for you. Items can be purchased multiple times, but after each item purchase, the item's cost will increase at an exponential rate, given by $C_n = C_0 \cdot \alpha^n$, where C_0 is the cost of the first item and C_n is the cost of item n. In the actual game, $\alpha = 1.15$. Because of the exponential cost growth of these items, item costs will grow faster than cookie generation rates can catch up, so cookie generation rates will not blow up exponentially over time.

In the actual game, there is no end condition. In this paper, we will analyze a variant of Cookie Clicker where the goal is to reach M cookies in the shortest amount of time possible. We will assume that the rate of cookies baked per second from clicking on the big cookie icon is 1. In this paper, I will use "Cookie Clicker" to refer to this specific variant of Cookie Clicker.

We solve Cookie Clicker for the 1-item case for all exponential growth rates. We then provide positive results regarding specific cases of the problem, and provide negative results about a generalized version and a discrete version of Cookie Clicker.

2 Basic Example: 1 Item Cookie Clicker

2.1 Model

Suppose you start with 0 cookies and your goal is to reach M cookies as quickly as possible. The only item available has initial cost y, increases your cookies per second by x (we can call this quantity the rate increase), and has cost gain $\alpha \geq 1$. You now seek to find the optimal sequence and timing of item purchases to achieve your goal. Given these constants, we define a game state as a tuple (c, n) where c is the number of cookies you have and n is the number of items you have. Note that your current status in the game is entirely described by the game state.

2.2 Useful tools

Claim 2.1. *If the next step of the optimal strategy involves buying an item at some point in the future, you should buy the item as soon as you can afford it.*

Proof. Suppose that from a given game state, a strategy involves buying the item t seconds after you can afford it. Let G denote the cookie generation rate at the current game state. The net change in game state after these t seconds is that you gained 1 item, and your cookie amount changed by $Gt - y'$.

Then consider the strategy that buys the item as soon as you can afford it and waits t seconds afterwards. At that time, the net change in game state after these t seconds is that you gained 1 item, and your cookie amount changed by $(G + x)t - y'$. Thus, this new strategy results in the exact same thing as the original strategy, except that it gains an extra tx cookies, which is strictly better. Thus, an optimal strategy that intends to buy an item must buy it as soon as it can be afforded. □

This claim tells us that the optimal strategy will always decide to wait until it can purchase an item and purchase it immediately, or it will wait until the target number of cookies M is reached. Thus, the problem really just involves jumping between game states in which you have 0 cookies and need to make a decision between waiting until the end or buying an item. This means that the only thing we need to keep track of to determine our game state is n, the number of items we have purchased. Thus, we can define $f(n)$ to be the minimum amount of time needed to reach M cookies from the game state $(0, n)$.

Claim 2.2. *If your current cookie generation rate is G, you should buy an item with cost y and rate increase x if and only if*

$$\frac{M}{y} \geq 1 + \frac{G}{x} \tag{1}$$

Proof. Suppose we are at a state where we have purchased n_0 items. Then, the optimal decision is either to purchase another item or to wait until we reach the goal without any more purchases.

In the first case, the time taken is equal to

$$\frac{y}{G} + f(n_0 + 1) \leq \frac{y}{G} + \frac{M}{G + x}$$

because a valid (but possibly not optimal) strategy from the state $(0, n_0 + 1)$ is to just wait it out from there.

In the second case, the time taken is equal to $\frac{M}{G}$.

Thus, if it is the case that

$$\frac{y}{G} + \frac{M}{G + x} \leq \frac{M}{G} \tag{2}$$

then we should go with the first strategy and purchase an item. Rearranging (2) gives (1).

This means that if (1) is satisfied, purchasing the item is better. Now we show that if (1) is not satisfied, then waiting is better. Suppose that (1) is not satisfied, so $\frac{M}{y} < 1 + \frac{G}{x}$. Then for any rate $G' > G$ and $y' \geq y$, the inequality $\frac{M}{y'} < 1 + \frac{G'}{x}$ still holds. Written in the form of (2), this inequality becomes $\frac{y}{G} + \frac{M}{G+x} > \frac{M}{G}$. Now, suppose that the optimal strategy from this point forward is to purchase k items for some $k > 0$ and then wait. Let y_i and G_i denote the price and cookie generation rate after i item purchases from this point forward, and note that $G_m = G_{m-1} + x$. Then the time taken to achieve this equals

$$\begin{aligned}
\frac{y}{G} + \frac{y_1}{G_1} + \cdots + \frac{y_{k-1}}{G_{k-1}} + \frac{y_k}{G_k} + \frac{M}{G_k + x} &> \frac{y}{G} + \frac{y_1}{G_1} + \cdots + \frac{y_{k-1}}{G_{k-1}} + \frac{M}{G_k} \\
&= \frac{y}{G} + \frac{y_1}{G_1} + \cdots + \frac{y_{k-1}}{G_{k-1}} + \frac{M}{G_{k-1} + x} \\
&> \frac{y}{G} + \frac{y_1}{G_1} + \cdots + \frac{y_{k-2}}{G_{k-2}} + \frac{M}{G_{k-2} + x} \\
&\cdots \\
&> \frac{y}{G} + \frac{M}{G + x} \\
&> \frac{M}{G}
\end{aligned}$$

Thus, if (1) is not satisfied, then the optimal strategy is to wait. This completes the if and only if statement's proof. □

2.3 Solution

Armed with the tools we developed in the previous section, we solve the 1-Item Cookie Clicker problem. Based on the results of the previous section, the optimal strategy is to purchase k items for some $k \geq 0$ as soon as each item becomes affordable and then wait until we reach M cookies. The total time that this takes is

$$\sum_{n=0}^{k-1} \frac{y \cdot \alpha^n}{1 + nx} + \frac{M}{1 + kx} \tag{3}$$

Using the results of **Claim 2.2**, we know that we should stop buying items when $\frac{M}{y} < 1 + \frac{G}{x}$. Here, $G = 1 + kx$ and $y = y \cdot \alpha^k$.

In the special case of $\alpha = 1$, which we call the *fixed-cost* case, the inequality becomes

$$\begin{aligned}
\frac{M}{y} &< 1 + \frac{1 + kx}{x} = 1 + k + \frac{1}{x} \\
\iff \frac{M}{y} - 1 - \frac{1}{x} &< k
\end{aligned}$$

so k is the smallest integer larger than $\frac{M}{y} - 1 - \frac{1}{x}$. In this case, the total time is equal to

$$\begin{aligned}
\sum_{n=0}^{k-1} \frac{y}{1+nx} + \frac{M}{1+kx} &= \frac{y}{x}\sum_{n=0}^{k-1} \frac{1}{1/x+n} + \frac{M}{1+kx} \\
&\approx \frac{y}{x}\sum_{n=0}^{k-1} \frac{1}{n} + \frac{M}{1+kx} \\
&\approx \frac{y}{x}\ln k + \frac{M}{1+kx} \\
&\approx \frac{y}{x}\ln \frac{M}{y} + \frac{M}{\frac{Mx}{y}} \\
&= \frac{y}{x}(\ln \frac{M}{y} + 1)
\end{aligned}$$

If $\alpha > 1$, the inequality then becomes

$$\begin{aligned}
\frac{M}{y \cdot \alpha^k} &< 1 + \frac{1+kx}{x} = 1 + k + \frac{1}{x} \\
\iff \frac{M}{y(1+k+\frac{1}{x})} &< \alpha^k \\
\iff \log_\alpha \frac{M}{y} - \log_\alpha (1+k+\frac{1}{x}) &< k
\end{aligned}$$

In most reasonable cases, the log term on the left hand side of the inequality is fairly small, so $k \approx \log_\alpha \frac{M}{y}$.

3 Positive Results

3.1 Fixed Cost Cookie Clicker for 2 Items

In this section, we analyze the cases where all the α's are equal to 1, which we call *Fixed Cost Cookie Clicker*. This is a natural starting point, as items should be fixed in price if there is enough supply.

In the 2-Item Cookie Clicker problem, our goal is to reach M cookies as quickly as we can, and the 2 items available are described by the tuples (x_1, y_1) and (x_2, y_2). These are defined analogously to the 1 item case. Given these constants, we define a game state as a tuple (c, n_1, n_2) where n_1 and n_2 are the number of item 1 and item 2 you have obtained respectively. We can apply the two claims from the 1 item case here too, so we know that the optimal strategy will jump between states where we have 0 cookies, and we can also use the same decision rule to determine whether or not purchasing an item is worth it at any given state. [FIX THIS - Actually, I don't think I've shown this yet - need to make sure that Claim 2.2 extends beyond 1-item case]. OK, using a different version of the claim (G is bounded by M/y ¿= 1 + G/x for the (x,y) pair that maximizes the lower bound.

Without loss of generality, we can assume that $y_2 > y_1$. We now solve this problem. Here, I claim that we can compute the optimal solution in polynomial time.

In this problem, we will also make the assumption that $\frac{x_2}{y_2} > \frac{x_1}{y_1}$. This is because if the reverse inequality held, then buying $\frac{y_2}{y_1}$ copies of item 1 gives a higher rate increase than buying a single instance of item 2, which means that it should never be optimal to buy item 2 if M is large enough (M must be large enough so that the effect of $\frac{y_2}{y_1}$ not being an integer is irrelevant in the long run).

Definition 3.1. *The efficiency score of an item of cost y and rate increase x when you have generation rate G is $\frac{y}{x} + \frac{y}{G}$.*

Claim 3.2. *You should always buy the item with the lower efficiency score.*

Let $T = \frac{y_2 - y_1}{\frac{y_1}{x_1} - \frac{y_2}{x_2}}$. Then, if $G < T$, you should never purchase item 2 followed by item 1, and if $G > T$, you should never purchase item 1 followed by item 2.

Proof. The efficiency score of an item basically dictates whether or not buying item 1 then item 2 is better than buying item 2 then item 1.

Suppose we have generation rate G. Then the cost of buying item 1 then item 2 is equal to $\frac{y_1}{G} + \frac{y_2}{G+x_1}$ and the cost of buying item 2 then item 1 is equal to $\frac{y_2}{G} + \frac{y_1}{G+x_2}$. If $G < T$, then we have that

$$\begin{aligned}
\frac{y_1}{G} + \frac{y_1}{x_1} &< \frac{y_2}{G} + \frac{y_2}{x_2} \\
\iff y_1\left(\frac{G+x_1}{x_1}\right) &< y_2\left(\frac{G+x_2}{x_2}\right) \\
\iff y_1\left(\frac{x_2}{G(G+x_2)}\right) &< y_2\left(\frac{x_1}{G(G+x_1)}\right) \\
\iff y_1\left(\frac{1}{G} - \frac{1}{G+x_2}\right) &< y_2\left(\frac{1}{G} - \frac{1}{G+x_1}\right) \\
\iff \frac{y_1}{G} + \frac{y_2}{G+x_1} &< \frac{y_2}{G} + \frac{y_1}{G+x_2}
\end{aligned}$$

On the other hand, if $G > T$, then the reverse is true.

□

Now, suppose that we have some optimal solution, which we can represent as a list of 1's and 2's, which represent which item to buy next. Now, we know that until the rate G reaches T, we will never have a 2 followed by a 1. Similarly, after the rate G passes T, we will never have a 1 followed by a 2. Thus, the final list is of the following form.

$$11\ldots1122\ldots2211\ldots11$$

Somewhere in the middle of the list of 2's, the generation rate reaches T.

Now, I will show that for large enough M, there will be no list of 1's at the end of the optimal solution.

Claim 3.3. *Let* $f(x_1, x_2, y_1, y_2) = \max\left(2, \frac{2}{x_1} \times \frac{y_1+y_2}{\frac{y_1}{x_1} - \frac{y_2}{x_2}}\right)$ *If* $M \geq (f+2)y_1$*, then the optimal solution will have no 1's at the end.*

Proof. Suppose for the sake of contradiction that there are k 1's at the end of the list representing the optimal solution for some $k > 0$. I will show that replacing the final 1 with a 2 results in a better solution, which disproves the optimality of the original solution.

Denote that the rate before purchasing the final 1 in the optimal solution as R.

The time it takes to buy the final 1 and then save up until the goal M is reached is equal to $\frac{y_1}{R} + \frac{M}{R+x_1}$. The time it takes to buy a 2 instead of the final 1 and then save up until the goal M is equal to $\frac{y_2}{R} + \frac{M}{R+x_2}$. We want to prove that

$$\frac{y_2}{R} + \frac{M}{R+x_2} < \frac{y_1}{R} + \frac{M}{R+x_1} \tag{4}$$

$$\iff \frac{M}{R+x_2} - \frac{M}{R+x_1} < \frac{y_1}{R} - \frac{y_2}{R} \tag{5}$$

Now, we know from **Claim 2.2** and the fact that the optimal solution bought the final "1" that

$$1 + \frac{R}{x_1} < \frac{M}{y_1} \iff \frac{M}{R+x_1} > \frac{y_1}{x_1}$$

Similarly, we know that since the optimal solution can not buy another "2" after the final "1" that

$$1 + \frac{R+x_1}{x_2} \geq \frac{M}{y_2} \iff \frac{M}{R+x_1+x_2} \leq \frac{y_2}{x_2}$$

Combining the above two equations, we end up with

$$\begin{aligned}
\frac{M}{R+x_1+x_2} - \frac{M}{R+x_1} &< \frac{y_2}{x_2} - \frac{y_1}{x_1} \\
\iff \frac{M}{R+x_2} - \frac{M}{R+x_1} &< \frac{y_2}{x_2} - \frac{y_1}{x_1} + \frac{M}{R+x_2} - \frac{M}{R+x_1+x_2} \\
&= \frac{y_2}{x_2} - \frac{y_1}{x_1} + \frac{Mx_1}{(R+x_2)(R+x_1+x_2)} \\
&< \frac{y_2}{x_2} - \frac{y_1}{x_1} + \frac{Mx_1}{R^2}
\end{aligned}$$

Thus, to prove (5), we just have to prove that

$$\begin{aligned}
\frac{y_2}{x_2} - \frac{y_1}{x_1} + \frac{Mx_1}{R^2} &< \frac{y_1}{R} - \frac{y_2}{R} \\
\iff \frac{Mx_1}{R^2} + \frac{y_2 - y_1}{R} &< \frac{y_1}{x_1} - \frac{y_2}{x_2}
\end{aligned}$$

Now note that because the optimal solution can not buy another "1" after the final "1,"

$$1 + \frac{R + x_1}{x_1} \geq \frac{M}{y_1} \iff R \geq \Big(\frac{M}{y_1} - 2\Big)x_1$$

Since $M \geq (f+2)y_1$ and $f \geq 2$, we can deduce that

$$R \geq \Big(\frac{M}{y_1} - 2\Big)x_1 = fx_1 \geq \frac{f+2}{2}x_1 = \frac{Mx_1}{2y_1}$$

Thus, $\frac{Mx_1}{R^2} = \frac{M}{R}\frac{x_1}{R} \leq \frac{2y_1}{x_1}\frac{x_1}{R} = \frac{2y_1}{R}$. Using this, all we have to prove now is that

$$\frac{2y_1}{R} + \frac{y_2 - y_1}{R} = \frac{y_1 + y_2}{R} < \frac{y_1}{x_1} - \frac{y_2}{x_2}$$
$$\iff \frac{y_1 + y_2}{\frac{y_1}{x_1} - \frac{y_2}{x_2}} < R$$

But this is true because we know

$$R \geq \frac{Mx_1}{2y_1} \geq \frac{(f+2)x_1}{2} > \frac{fx_1}{2} > \frac{y_1 + y_2}{\frac{y_1}{x_1} - \frac{y_2}{x_2}}$$

□

Thus, we have shown that for large enough M, the optimal solution will be of the form

$$11\ldots1122\ldots22$$

where the 1's only appear if the total generation rate at that point is less than the threshold T. We can experimentally verify that the point at which the optimal solution transitions from 1's to 2's is not exactly T

To solve this problem, we just have to find the optimal number of 1's to buy before transitioning to 2's and solving the 1-item cookie clicker game, which is simple. Finding this optimal number involves maximizing a function of a single variable (the number of 1's to buy before transitioning to 2's), which should not be that hard.

3.2 Fixed Cost Cookie Clicker for n Items

A natural follow-up is to extend this problem from 2 items to n items. So far, no progress has been made on a strongly polynomial time solution, though a weakly polynomial dynamic programming solution exists. Experimentally, it appears that

3.2.1 Dynamic Programming Solution

3.2.2 Local Optimizations

No solution obtained. Can use DP to solve. We tried to use LOCAL OPTIMIZATIONS. LISTS RESULTS HERE. There are local maxes.

3.3 Increasing Cost Cookie Clicker (n-item)

From this point forward, we assume that the cost increase rates $\alpha > 1$. This is how the original Cookie Clicker game works, and is also a reasonable assumption for cases where the supply of each item is limited, and thus items would increase in price when more of them are purchased.

3.3.1 Natural Greedy Solution

One greedy solution that arises naturally in normal gameplay involves buying the item that has the highest rate increase to cost ratio $\frac{x_i}{y_i}$ if it is better to buy that item than to wait. This is the calculation that most human players do when playing the game unaided. For the 2 item case, given most reasonable setting of the parameters, this approach actually performs fairly well. However, for certain settings of the parameters, this approach can be quite bad. For example, take $M = 10000, (x_1, y_1, \alpha_1) = (1, 10, 2), (x_2, y_2, \alpha_2) = (10000, 9999, 2)$.

3.3.2 Frozen Cookie Greedy Solution

As we saw in the analysis of the fixed cost case, the efficiency score was a helpful metric to determine which item to buy. I implemented this greedy solution for the 2 item case and compared it with the optimal solution obtained from the dynamic programming approach.

3.3.3 Approximation Ratio

We derive an approximation ratio of $1 + \epsilon$ for the case where $\alpha_1, \alpha_2 \geq \frac{4}{3}$ for sufficiently large M.

The main idea is to use two facts (for large enough M):

1. Beyond any point that's not that close to M, you'll want to purchase another copy of item 1 and another copy of item 2

2. When G is big enough, α is big enough such that the most efficient item is locally and globally optimal

Intuitively, the second point should be true if the first point is. If, WLOG, it is currently better to buy item 1 then item 2 rather than item 2 then item 1, then it will not be better to buy some number of item 2's before buying item 1 because you could then swap the order in which you buy that item 1 and the previous item 2's. Each purchase of item 2 makes item 2 more expensive, so if 1 then 2 is better than 2 then 1, then it should be true that, for example, 212 is better than 221. In general, the claim is that

$$22\ldots221 \text{ is worse than } 22\ldots212\ldots \text{ is worse than } 122\ldots2$$

The reason this logic might break down is that even though item 2 gets more expensive (and thus less efficient) with more purchases, the generation rate also goes up, which could actually improve the efficiency score of item 2 relative to item 1.

For this problem, we will consider the following setup. Let $\alpha_1, \alpha_2 \geq \frac{4}{3}$ and

Claim 3.4. *Let $q_2 = \frac{G}{x_2}$. Suppose that at generation rate G, y_1 has a lower efficiency score than y_2. Then the next item that should be purchased is item 1 if $q_2^2 + 2q_2 \geq \frac{1}{\alpha_2 - 1}$.*

Proof. If item 1 is more efficient than item 2 at generation rate G, then

$$\frac{y_1}{x_1} + \frac{y_1}{G} \leq \frac{y_2}{x_2} + \frac{y_2}{G}$$

We want to find conditions such that the correct next item to purchase in the optimal solution is **NOT** item 2. Suppose for the sake of contradiction that the optimal solution is to buy some number of item 2's and then item 1.

First, we know that $12 > 21$. [I should define some notation for this]. Now, we will find conditions where $212 > 221$.

To show that $212 > 221$, we need to prove that

$$\frac{y_1}{x_1} + \frac{y_1}{G + x_2} \leq \alpha_2\left(\frac{y_2}{x_2} + \frac{y_2}{G + x_2}\right)$$

□

We know that
$$\frac{y_1}{x_1} + \frac{y_1}{G + x_2} < \frac{y_1}{x_1} + \frac{y_1}{G} \leq \frac{y_2}{x_2} + \frac{y_2}{G}$$

so we just have to show that

$$\begin{aligned}
\frac{y_2}{x_2} + \frac{y_2}{G} &\leq \alpha_2\Big(\frac{y_2}{x_2} + \frac{y_2}{G+x_2}\Big) \\
\iff \frac{y_2}{G} &\leq (\alpha_2 - 1)\frac{y_2}{x_2} + \frac{\alpha_2 y_2}{G+x_2} \\
\iff \frac{1}{x_2 q_2} &\leq (\alpha_2 - 1)\frac{1}{x_2} + \frac{\alpha_2}{x_2 q_2 + x_2} \\
\iff q_2 + 1 &\leq (\alpha_2 - 1)q_2(q_2+1) + \alpha_2 q_2 \\
\iff 1 &\leq (\alpha_2 - 1)(q_2^2 + 2q_2) \\
\iff \frac{1}{\alpha_2 - 1} &\leq q_2^2 + 2q_2
\end{aligned}$$

This is the original assumption. Thus, we know that $212 > 221$ if the original condition holds. We also know that $122 > 212$, so the optimal solution from rate G can not start with a 21 sequence or a 221 sequence.

Now, I claim that this holds for any string of 2's in the beginning - that is, $22\ldots212 > 22\ldots221$. This is apparent because at the point where the 12 needs to be compared to the 21, the generation rate there $G' > G = q_2x_2$,

CONTINUE WORKING HERE. FINISH PROVING THE APPROXIMATION RATIO.

However, experimental results (of which I have recorded many) show that the Frozen Cookie greedy solution performs exceptionally well relative to the optimal solution, and almost always matches the optimal solution, except possibly at the very beginning of the problem. Even then, at most a few decisions are ordered differently, and after the initial differences the two strategies match up exactly.

Some ideas for finding an approximation ratio - it seems to be the case that the final number of each item purchased in both the true optimal solution and the frozen cookie greedy solution are almost always the same.

One greedy algorithm that could be less efficient but would be easier to analyze would be to purchase 1 of each item as quickly as possible and then follow the Frozen Cookie Greedy Solution from there.

4 Negative Results

4.1 General Cookie Clicker

We will now define the General Cookie Clicker problem. The inputs to this problem are

- k, the number of cookies you start out with
- Vectors X, Y, and α, where each triple (x_i, y_i, α_i) represents the (rate gain, initial cost, cost gain) of each item
- r, the initial generation rate

- M, the target number of cookies

The goal of this game is to find the optimal order of items to purchase to reach the goal M as quickly as possible. This game is the original cookie clicker game with the added flexibility of starting with any number of cookies in the bank and a variable initial generation rate.

4.1.1 Dynamic Programming Solution

First, let us solve the problem for the case with 2 items.

Recall that your state in the game is completely described by the tuple (n_1, n_2). Thus, we can use a dynamic programming approach to find the optimal solution.

Note that it will never be worth it to buy an item if the item costs more than the goal M. This gives us upper bounds on n_1 and n_2, namely, $n_i < \log_{\alpha_i} \frac{M}{y_i} + 1$. Thus, we define

$DP[a][b] :=$ the minimum time it takes to reach M from the state (a,b).

Let the cookie generation rate at state (a, b) be represented by $g_{ab} = 1 + ax_1 + bx_2$. From the state (a, b), the optimal strategy is one of the following three choices - waiting, buying item 1, or buying item 2. We can then derive that we have the recurrence

$$DP[a][b] = \min\left(\frac{M}{g_{ab}}, \frac{y_1 \cdot \alpha_1^a}{g_{ab}} + DP[a+1, b], \frac{y_2 \cdot \alpha_2^b}{g_{ab}} + DP[a, b+1]\right) \tag{6}$$

If A and B are the upper bounds for n_1 and n_2 respectively, then we can initialize $DP[A][b]$ and $DP[a][B]$ for any a and b in the proper range by solving the 1 item problem in those cases. We can then use the recurrence to fill out the rest of the dynamic programming table. Finally, our answer is $DP[0][0]$.

Initializing the boundaries of the table by solving the 1 item problem takes $O(A + B)$ time because filling in each individual entry just involves checking the constraints of Claim 2.2, which takes $O(1)$ time. Then, filling out the rest of the AxB table takes $O(AB)$ time, so the total runtime is $O(AB)$.

In fact, the dynamic programming approach can easily be extended to the k item problem. As before, one can note that the game state of the k item problem is described entirely by the k-tuple $(n_1, n_2, \ldots, n_k)$, where n_i is the number of item i that you have purchased. We can use the same upper bound $n_i < \log_{\alpha_i}(\frac{M}{y_i}) + 1$. Let $N_i = \log_{\alpha_i}(\frac{M}{y_i}) + 1$ denote the upper bounds for each n_i.

We can define

$DP[(n_1, n_2, \ldots, n_k)] :=$ the minimum time it takes to reach M from state $(n_1, n_2, \ldots, n_k)$

Then, filling in any square in the grid involves checking the solutions the adjacent squares and doing an $O(1)$ computation for each adjacent square. In total, this is at most $O(k)$ time. The only square we need to initialize is the corner $DP[(N_1, N_2, \ldots, N_k)]$. Thus, the total time complexity of this program is $O(k \prod_{i=1}^{k} N_i) = O(kN_{max}^k)$.

4.1.2 NP-hardness of the General Cookie Clicker problem

We will now define the General Cookie Clicker problem and prove that it is NP-hard. The inputs to this problem are

- k, the number of cookies you start out with
- Vectors X, Y, and α, where each triple (x_i, y_i, α_i) represents the (rate gain, initial cost, cost gain) of each item
- r, the initial generation rate
- M, the target number of cookies

The goal of this game is to find the optimal order of items to purchase to reach the goal M as quickly as possible. This game is the original cookie clicker game with the added flexibility of starting with any number of cookies in the bank and a variable initial generation rate.

Theorem 4.1. *General Cookie Clicker is NP-Hard.*

We will use a reduction from 2-Partition to prove the desired result.

Suppose we are given an instance of 2-Partition in the form of a set of integers $(a_1, a_2 \dots a_n)$ such that

$$\sum_{i=1}^{n} a_i = 2B$$

We will construct an instance of General Cookie Clicker such that solving General Cookie Clicker will solve 2-Partition.

To do so, we can construct an instance of General Cookie Clicker with the inputs set as follows.

- $k = nA + B$
- $(x_i, y_i, \alpha_i) = (a_i + A, a_i + A, \infty)$ for $1 \leq i \leq n$
- $(x_i, y_i, \alpha_i) = (A, A, \infty)$ for $n + 1 \leq i \leq 2n$
- $r = 0$
- $M = nA + B + 1$

[Would this be more impressive if we proved it for $r > 0$ or specifically for $r = 1$, as before? The same proof with some extra math should work]

Here, A is just some big number (for example, $A = 1000B$). The fact that $\alpha_i = \infty$ means that we can buy at most 1 of each item. I will now show the following equivalence.

Claim 4.2. *A solution to 2-Partition exists if and only if the minimum amount of time it takes to reach M in the corresponding General Cookie Clicker problem is $\frac{M}{nA+B}$.*

Proof. First, suppose that a 2-Partition solution exists. That means we can choose some set of the integers $(a_1, a_2 \dots a_n)$ such that they sum to B. Equivalently, this means we can use our initial $k = nA+B$ cookies to buy n total items for a price of $nA+B$ at the very beginning of the game. We then wait until we have M cookies. The total amount of time it takes to reach M using this strategy is $\frac{M}{nA+B}$.

Next, I will show that there is no faster way to reach M cookies. Using **Claim 2.1**, we know that for any optimal strategy for General Cookie Clicker, the strategy will buy items as soon as it can up until a point, and then it will stop buying items and wait until the goal is reached. We now consider two cases.

Case 1: The strategy ends at a rate less than $nA + B$

If the strategy ends at a rate of $nA + B - j$ for $j > 0$, then it must have spent $nA + B - j$ purchasing items at $t = 0$ and then waited from that point forward. The total time that this strategy takes is $\frac{M-j}{nA+B-j}$. It's easy enough to verify that

$$\begin{aligned} \frac{M}{nA+B} < \frac{M-j}{nA+B-j} &\iff -jm < -j(nA+B) \\ &\iff nA+B < M \end{aligned}$$

which is true.

Case 2: The strategy ends at a rate greater than $nA + B$.

First, I claim that the strategy can buy at most n items. This is because after buying n items, there will not be enough cookies left over to purchase the next item right away. The General Cookie Clicker problem then reduces to the original Cookie Clicker problem where you have no cookies at the start. We can then recall from **Claim 2.2** that if you have generation rate G, it is only worth it to buy an item with rate increase x and cost y if

$$\begin{aligned} \frac{M}{y} &\geq 1 + \frac{G}{x} \\ \iff \frac{Mx}{y} &- x \geq G \end{aligned}$$

Since $x_i = y_i$ for all i, this becomes

$$M - x_i \geq G \tag{7}$$

After purchasing n items, $G \geq nA$. Then $G + x_i \geq G + A \geq (n+1)A > M$, so it is no longer worth it to buy any items after purchasing n items.

Furthermore, suppose that we have purchased $n - 1$ items. Then equation (6) tells us that item i is only worth purchasing if $M \geq G + x_i$. Thus, the final rate after purchasing the n^{th} item, $G + x_i$, is upper bounded by $M = nA + B + 1$. Since we are currently considering the case where the final rate is greater than $nA + B$, the only possible final rate for this case is then $nA + B + 1$.

Now, we have reduced this case to the specific scenario where $n - 1$ items are purchased at time $t = 0$, giving a rate of $(n-1)A + c$ for some $0 \le c \le 2B$. Then, the strategy purchases another item as soon as it can, ending up at a rate of $nA + B + 1$. The amount of time this strategy takes is

$$\frac{1}{(n-1)A + c} + \frac{M}{nA + B + 1} \geq \frac{1}{(n-1)A + 2B} + \frac{M}{nA + B + 1}$$

We want to check that the quantity on the RHS is greater than $\frac{M}{nA+B}$. Indeed

$$\begin{aligned} \frac{1}{(n-1)A + 2B} + \frac{M}{nA + B + 1} &> \frac{M}{nA + B} \\ \iff \frac{1}{(n-1)A + 2B} &> \frac{M}{(nA + B)(nA + B + 1)} \\ \iff \frac{1}{(n-1)A + 2B} &> \frac{1}{nA + B} \\ \iff nA + B &> (n-1)A + 2B \\ \iff A &> B \end{aligned}$$

which is true. Thus, any strategy that ends at a rate greater than $nA + B$ will also get to M slower than the proposed optimal method using the partition solution. The "if" direction of **Claim 5.2** is proven.

The "only if" direction of **Claim 5.2** is easy to prove using the same results. If we are given a solution to General Cookie Clicker that reaches the goal in $\frac{M}{nA+B}$ time, then our analysis for the "if" direction's proof shows that the final rate upon reaching the goal must have been $nA + B$. Since the cost and the rate of all items are the same, this means that we spent $nA + B$ on items, which means all of the money was spent at $t = 0$. The only way we could have spent exactly $nA + B$ at time $t = 0$ is if there existed a solution to 2-Partition, as desired.

□

Now, we can prove **Theorem 5.1**.

Proof. We have shown that given an instance of 2-Partition, which is NP-complete, we can construct an instance of General Cookie Clicker in polynomial time such that being able to solve General Cookie Clicker means being able to solve 2-Partition. Thus, General Cookie Clicker itself is NP-hard. □

4.2 Discrete Cookie Clicker

This is a variant of Cookie Clicker in discrete time, whereas we have been analyzing it in continuous time. This is the model of Cookie Clicker discussed by Jason Lynch and Mikhail Rudoy, which has also been proven to be NP-Hard.

Theorem 4.3. *Discrete Cookie Clicker is NP-hard*

5 Code

For a python implementation of both the DP solution and the greedy solution to the General Cookie Clicker problem, see here: https://github.com/kaixiao/Cookie-Clicker

6 Conclusion

I will add this later.