Time Spent: 2 hours

Collaborators and Resources: Talked with Nathaniel MacArthur-Warner about the divide-and-conquer method

---

Bonus Problem

Here we're given a situation in which we have a lot of restrictions- namely, we can make intraset comparisons neither within the group of bowls nor within the group of Goldilocks. Thus, we need to be able to make a bunch of recursive calls which will effectively generate a tree of groups of ever-decreasing size which eventually culminates in our desired result. The nature of this result is a little difficult at first to prove . . .

1. Given a set of Goldilocks $\{g_1, ..., g_n\}$, temperature preferences t_k for each g_k denoted by the set $\{t_1, ..., t_n\}$ and additionally the bowls of porridge $\{b_1, ..., b_n\}$
2. Initialize the set of desired pairings S = $\emptyset$
3. Define the function Pairings(Goldilocks, Bowls):
   (a) If all proper pairings are found, break
   (b) Otherwise
      i. Initialize four empty sets $Goldi_{cold}$, $Goldi_{hot}$, $Bowls_{cold}$ and $Bowls_{hot}$
      ii. For each bowl $b_k \in \{b_1, ..., b_n\}$
         A. Give b_k to g_1.
         B. If t_k ¡ b_k (that is, the porridge is too hot) then place the bowl b_k into $Bowls_{hot}$.
         C. If t_k ¿ b_k (that is, the porridge is too cold) then put the bowl b_k into the set $Bowls_{cold}$
         D. Otherwise, if they're equal then add the pair (g_1, b_k) to the set of desired pairings S. Remove g_1 and b_k from their respective sets.

E. Now that we have constructed two sets of bowls which are either too hot or too cold for our selected Goldilocks, we need to do a second testing.

iii. For each Goldilocks $g_k \in \{g_1, ..., g_n\}$

A. Take the bowl from the most recently discovered pair $(g_1, b_k) \in$ the desired pairings S and give it to g_k

B. During this second tasting, if t_k ¡ b_k (the porridge is too hot) then place the Goldilocks g_k into the set $Goldi_{hot}$

C. Otherwise, if t_k ¿ b_k (the porridge is too cold) then place the Goldilocks g_k into the set $Goldi_{cold}$.

D. (None of the remaining Goldilocks are going to find this bowl right because there is a bijective function between the set of Goldilocks and bowls)

(c) Return Pairings($Goldi_{hot}, Bowls_{cold}$)

(d) Return Pairings($Goldi_{cold}, Bowls_{hot}$)

Claim 1. Within the constraints given (only trials are permitted) the algorithm correctly matches the bowls to the Goldilocks in O($n$log$n$) time.

Proof. The argument for the runtime is as follows. Since we're making two recursive calls within the function $\implies$ b = 2, and dividing the Goldilocks and Bowls sets into two halves $\implies$ a = 2, case (i) of the Master Theorem justifies saying that Since f(n) = $\theta(n^{log_2 2})$, T = $\theta(n^{log_2 2}$ = θ(n) $\implies$ T(n) $\leq$ aT($\frac{n}{b}$ + f(n) = 2T($\frac{n}{2}$ + cn = $\theta(nlogn)$. Furthermore, bowls which were too hot for g_1 are going to be preferred by the Goldilocks who thought that the bowl b_k from the pair (g_1, b_k) was too cold. So we know that the algorithm is going to work and based on Master Theorem 1 we are able to demonstrate that its runtime (using only trials) is O($nlogn$). □