Daniel Leeds, 15-212 R09, October 24, 2007

Basic Lazy Lists:

```
datatype 'a lazyList = Cons of 'a lazyList * (unit -> 'a lazyList)

fun nats n = Cons (n, fn () => nats (n+1))

fun Map f (Cons(x, y)) = Cons( f x, fn () => Map f (y ()))


(* given L1 and L2 are sorted to be ascending lazy lists,
 * Merge L1 L2 returns a sorted lazy list containing all
 * the elements of L1 and of L2 *)
fun Merge (Cons(x,f)) (Cons(y,g)) =
```

Factorial Lists:

```
fun factgen n = Cons ( n, fn () => Map (fn x => n*x) (factgen (n+1)) )

(* What does this do? *)
fun lotsOfFacts n = Cons ( factgen n, fn () => lotsOfFacts (n+1))
```

Taxicab Numbers:

(According to Wikipedia…) The *n*th taxicab number, typically denoted Ta(*n*), is defined as the smallest number that can be expressed as a sum of two positive cubes in *n* distinct ways, up to the order of summands.

$Ta(1) = 2 = 1^3+1^3$
$Ta(2) = 1729 = 1^3 + 12^3 = 9^3 + 10^3$
$Ta(3) = 87539319 = 167^3 + 436^3 = 228^3 + 423^3 = 255^3 + 414^3$

Exercise: Write a function (really, a set of functions) to find Ta(n).