CSE 350

Advanced Data Structures

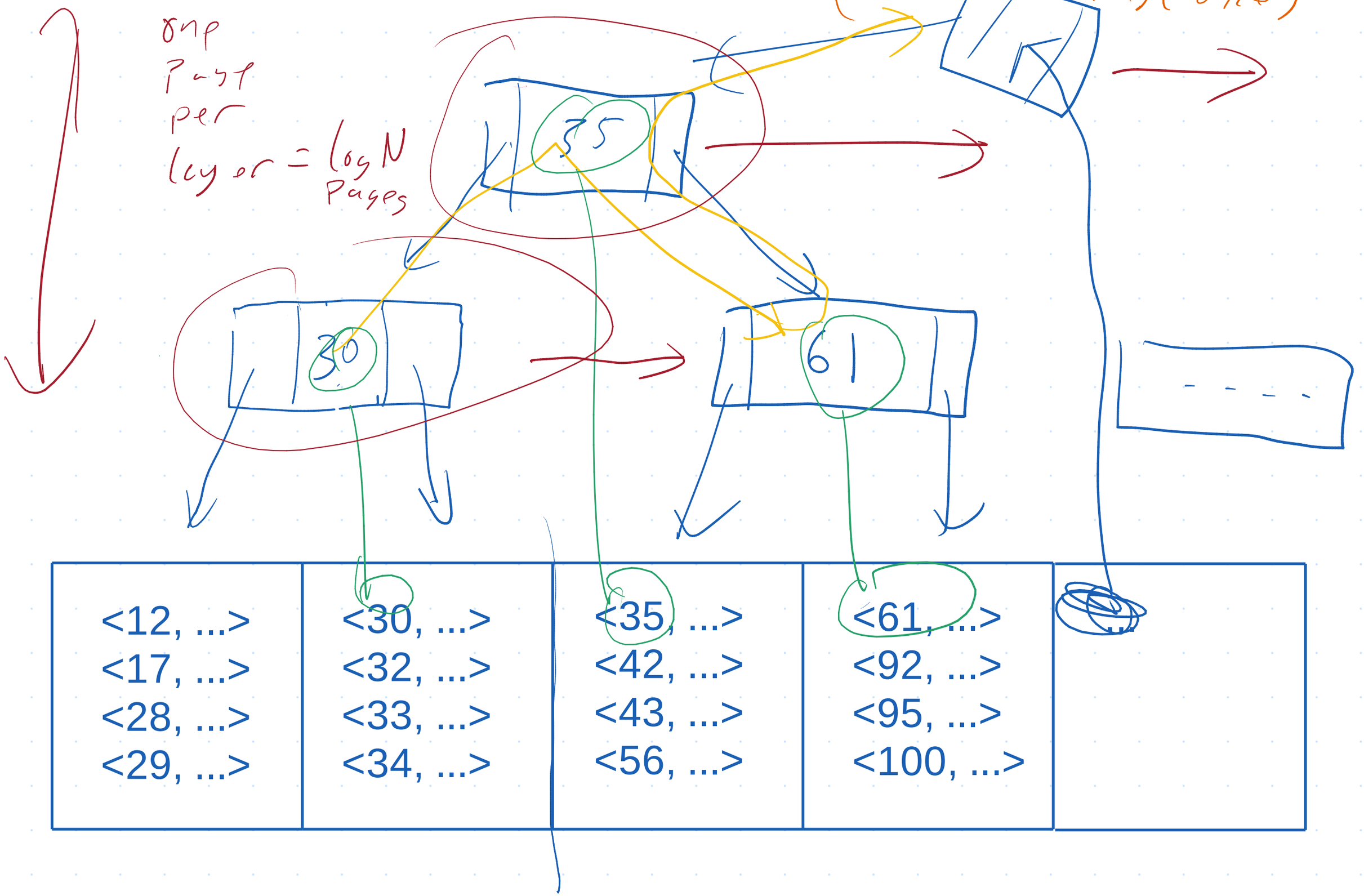
Topic 13: Indexing

Bulk loading trees

P_0	P_1	P_2	P_3	
<29, ...> <28, ...> <12, ...> <95, ...>	<35, ...> <43, ...> <30, ...> <92, ...>	<100, ...> <42, ...> <56, ...> <33, ...>	<17, ...> <61, ...> <32, ...> <34, ...>	...

O(N) reads (each page once) = O(N) IOs
O(N) writes (each B+Tree page once)

One page per layer = log N Pages



<12, ...> <17, ...> <28, ...> <29, ...>	<30, ...> <32, ...> <33, ...> <34, ...>	<35, ...> <42, ...> <43, ...> <56, ...>	<61, ...> <92, ...> <95, ...> <100, ...>	

Secondary Indexes

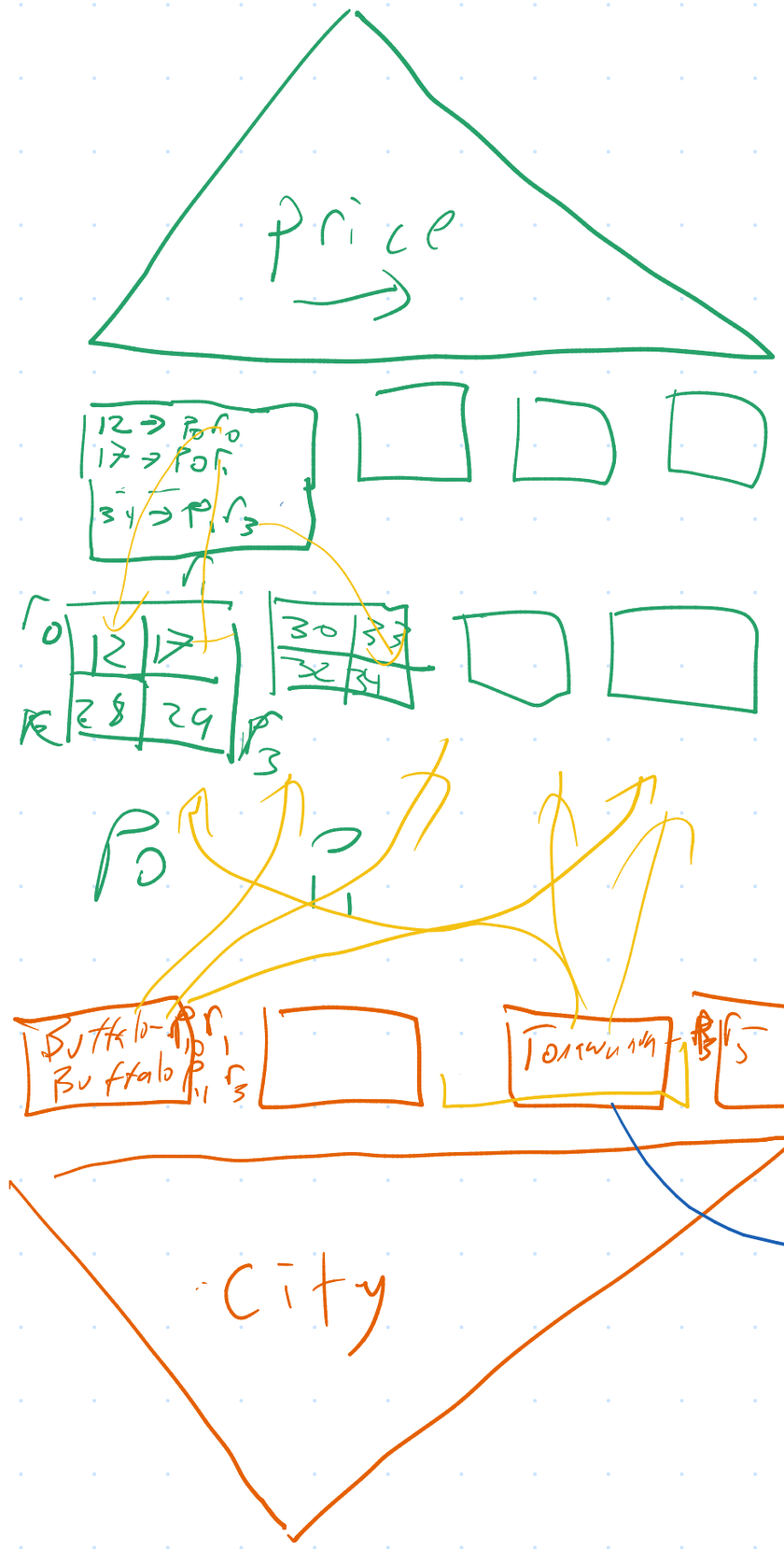
Leaf indirection

Permits
- ↳ Price Range
- ↳ City
- ↳ Type

```
SELECT COUNT(*)
FROM PERMITS
WHERE Price ≥ $1000
```

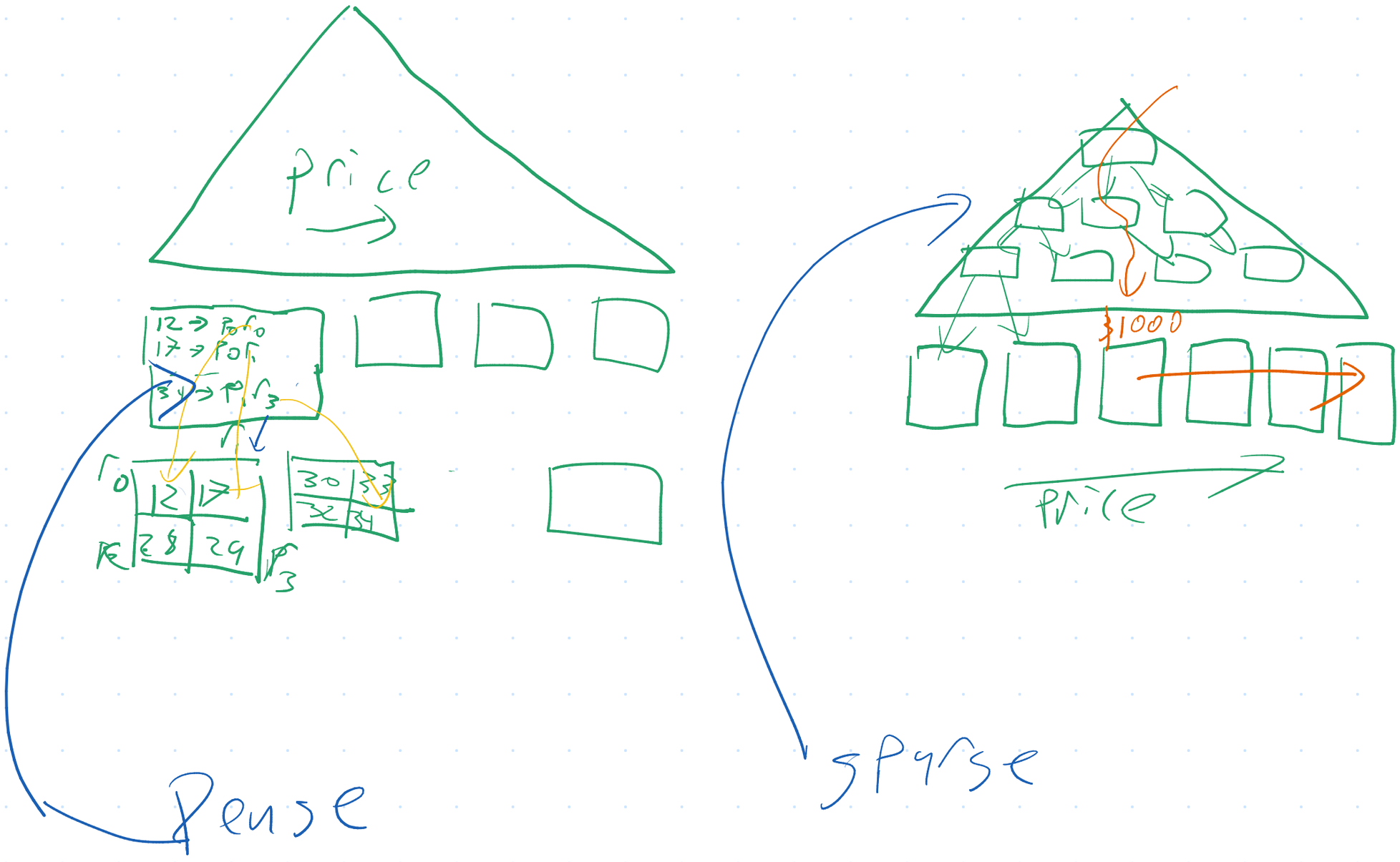
```
SELECT COUNT(*)
FROM PERMITS
WHERE City = Tongwarda
```

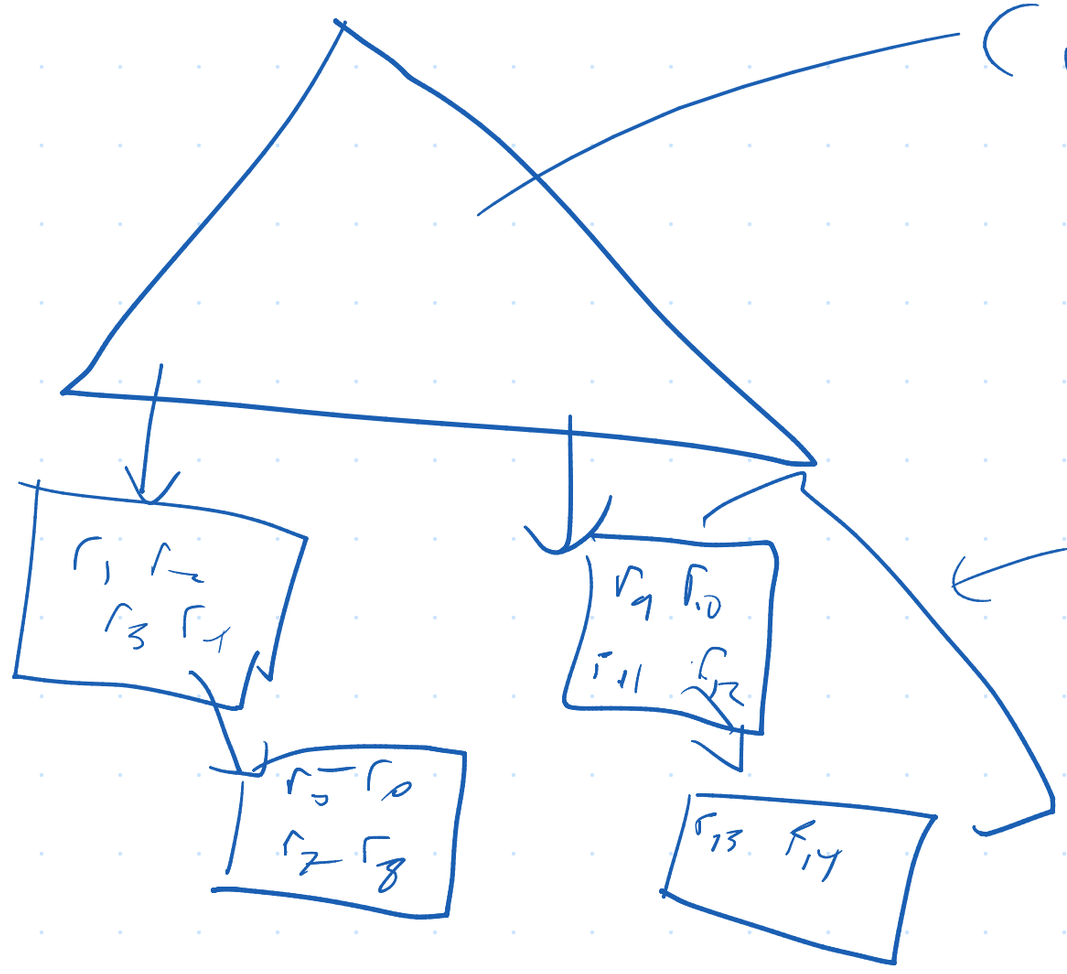
Clustered


price
directory pages
leaf pages
data pages
B+ Tree
K → tuple
K → address
Unclustered
city
SELECT Count (*)
WHERE City = Tamarindo
↳ Index has one entry per record already
Data not needed!

Clustered vs Unclustered

Dense, Sparse, Coarse-grained Indexes

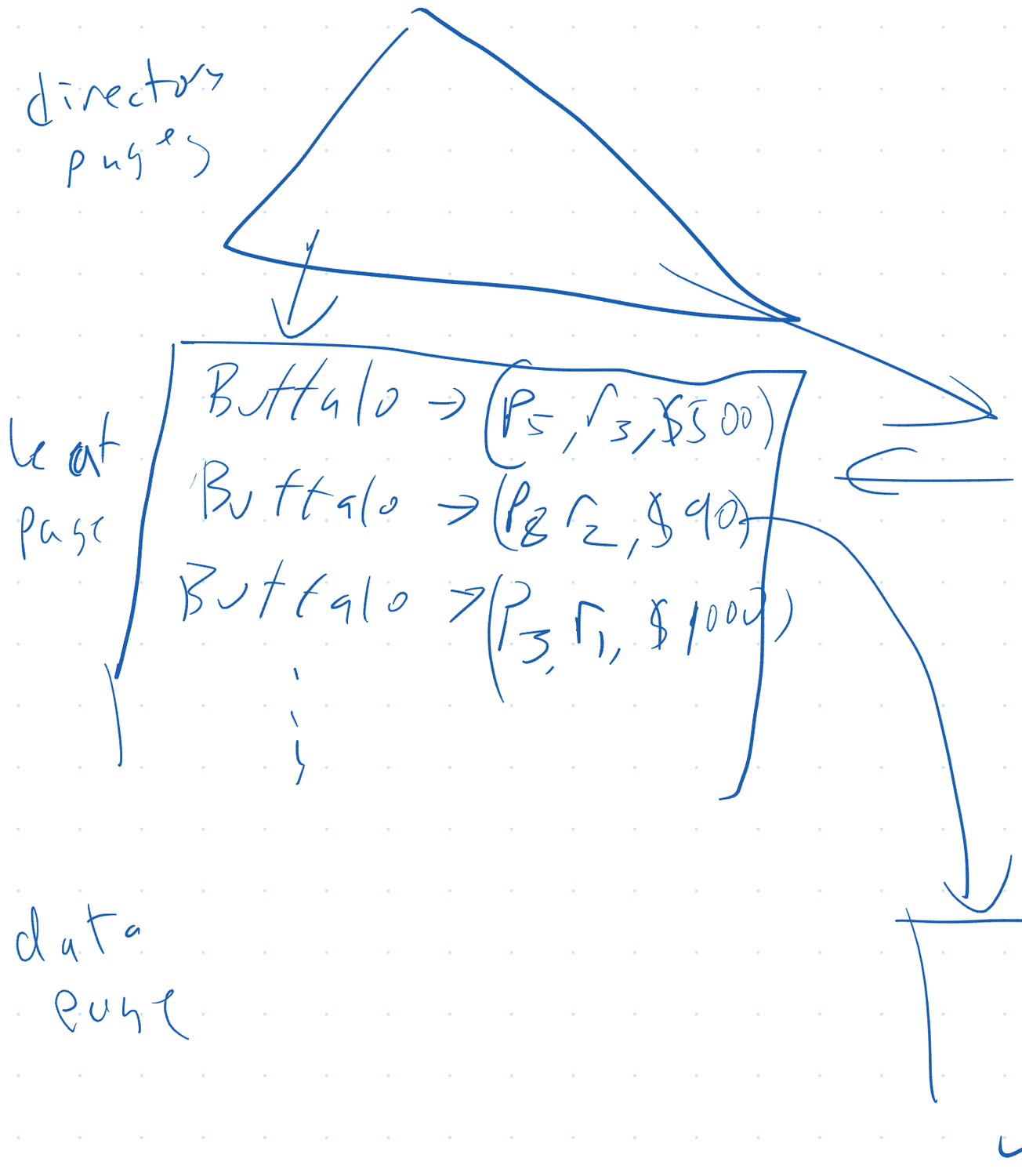




Useful

- much duplication
- queries looking for large ranges

Covering Indexes




```
SELECT SUM(PRICE)
FROM Permits
WHERE City = Buffalo
```

city, price are covered

covered attrs in query can be answered only w/ index w/o data pages

Composite Indexes

(City, Price)

Index w/ multiple Attrs as key

by lexical sort on City, Price in a B+Tree

useful for filtering on

- =City, ≥city
- =City =Price
- =city ≥Price

But not on

just price