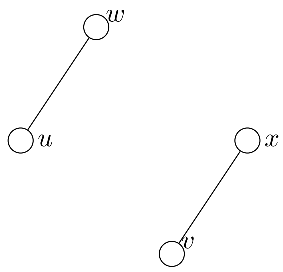
Let P be a set of n points in $\mathbb{R}^d$, where d is a constant. Denote by T the quadtree of P. In the lecture, we proved that our s-WSPD algorithm computes an s-WSPD of P with $O(s^d \cdot n \cdot h)$ pairs, where h is the height of T. Now, apply the same algorithm on the *compressed* quadtree tree T_{com} of P. In this exercise, you will prove that the algorithm produces an s-WSPD of $O(s^d \cdot n)$ pairs.

We will apply the same charging strategy as introduced in the lecture. Every time the algorithm generates $\{u, v\}$ from $\{w, v\}$ by splitting w (i.e., w is the parent of u in T_{com}), we charge the pair $\{u, v\}$ on w.



Solve the following problems.

Problem 1. For each node z in T_{com}, we use $level(z)$ to denote the level of z in the original quadtree T.
Suppose the algorithm generates $\{u, v\}$ from $\{w, v\}$ by splitting w (i.e., w is the parent of u in T_{com}), and x is the parent of v, prove: $level(v) \geq level(w) \geq level(x)$.
Remark: Recall that if a node is at level ℓ of T, the node corresponds to a box with side length $1/2^\ell$ on each dimension. Essentially, you need to prove that the box of v is no larger than that of w, which is in turn is no larger than that of x.

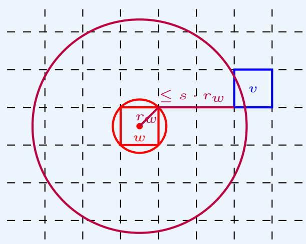
Proof. Since x is the parent of v, $level(x) \leq level(v)$.

The algorithm always split the larger node in a pair. A larger node corresponds to a node with larger side length. Thus, $level(v)$ is no less than $level(w)$. □

Problem 2. Fix a node w in T_{com} and a child u of w. Prove: there are $O(s^d)$ nodes v in T_{com} satisfying

1. $level(w) = level(v)$;
2. w is charged for the pair $\{u, v\}$.

Proof. w is charged for the pair $\{u, v\}$ if $level(w) \geq level(v)$ and w, v is not well separated.



There are in total $O(s^d)$ nodes v with the required property, given by the packing lemma. □

Problem 3. Fix a node w in T_{com} and a child u of w. Prove: there are $O(s^d)$ nodes v in T_{com} satisfying

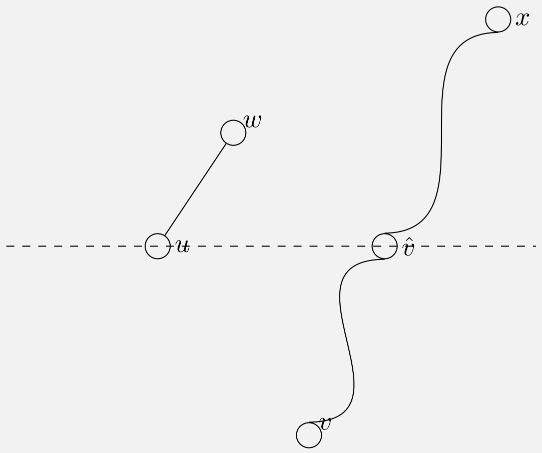
1. $level(w) = level(x)$ where x is the parent of v in T_{com};
2. w is charged for the pair $\{u, v\}$.

Proof. Consider the nodes described by Problem 2. There are at most $O(s^d)$ parents x for these nodes which can possibly satisfy the requirement. $\square$

Problem 4. Fix a node w in T_{com} and a child u of w. Now, we are proving: there are $O(s^d)$ nodes v (which is then described by the set S) in T_{com} satisfying

1. $level(v) > level(w) > level(x)$ where x is the parent of v in T_{com};
2. w is charged for the pair $\{u, v\}$.

Consider any node $v \in S$ and let x be the parent of v in T_{com}. Identify the node in T (the original quadtree) at level $level(w)$ on the path from x to v in T. We will refer to $\hat{v}$ the anchor node of v with respect to w. Note that $\hat{v}$ has only a single child and does not exist in T_{com} (i.e. $\hat{v}$ is removed by compression).



Prove: the node in S have distinct anchor nodes with respect to w.
Hint: Which node on the path from x to v in T have only one child?

Proof. Suppose that the node in S are having non-distinct anchor nodes with respect to w. Then, node $\hat{v}$ has more than one descendent, which is a contradiction. $\square$

Problem 5. Fix a node w in T_{com} and a child u of w. We are proving: there are $O(s^d)$ nodes v (which is then described by the set S) in T_{com} satisfying

1. $level(v) > level(w) > level(x)$ where x is the parent of v in T_{com};
2. w is charged for the pair $\{u, v\}$.

Prove: $|S| = O(s^d)$.
Hint: Apply the Packing Lemma to bound the number of anchor nodes.

Proof. Recall the proof in Problem 2 counting the number of nodes v. Now, counting the number of nodes $\hat{v}$ is the similar setting to the one in Problem 2. $\square$

Problem 6. Prove: Each node w of T_{com} can be charged only $O(s^d)$ times.

Proof. All cases are identified by Problem 2 - Problem 5.

w has at most 4 child. The number of possible nodes v forming a pair with u such that w is charged by the algorithm is bounded by $O(s^d)$.

Thus, each node in T_{com} can be charged only $O(s^d)$ times. $\square$