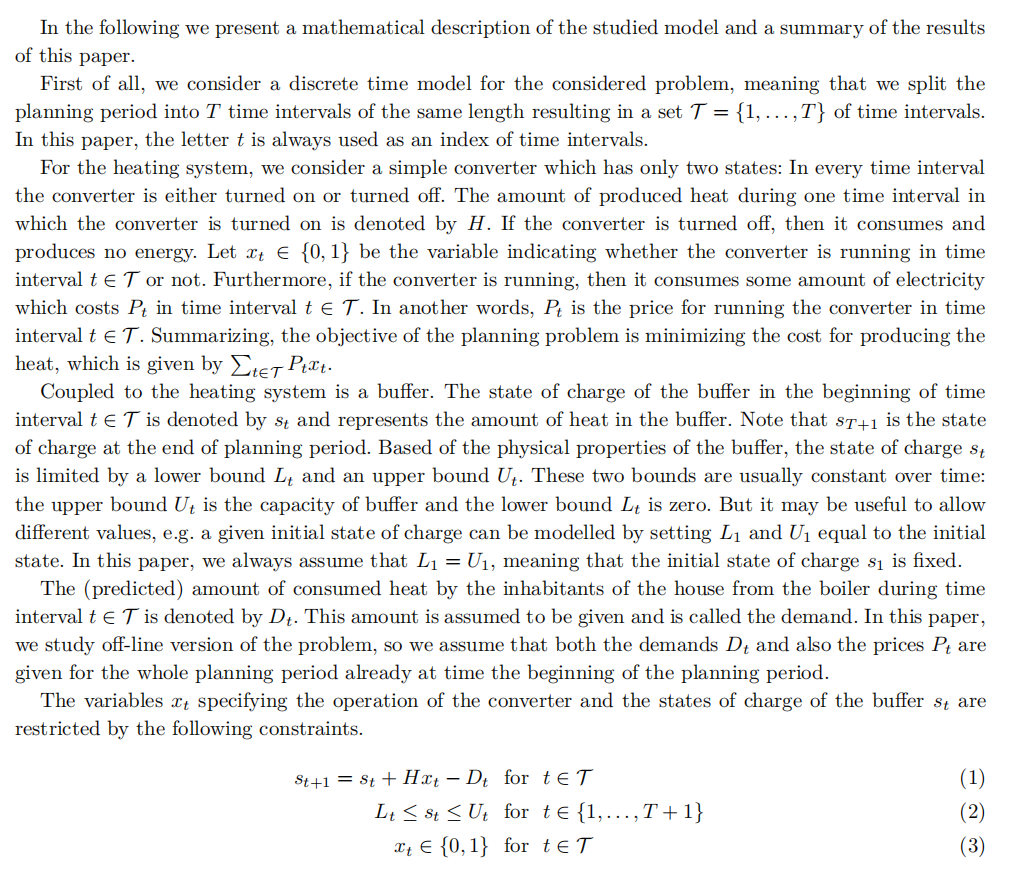
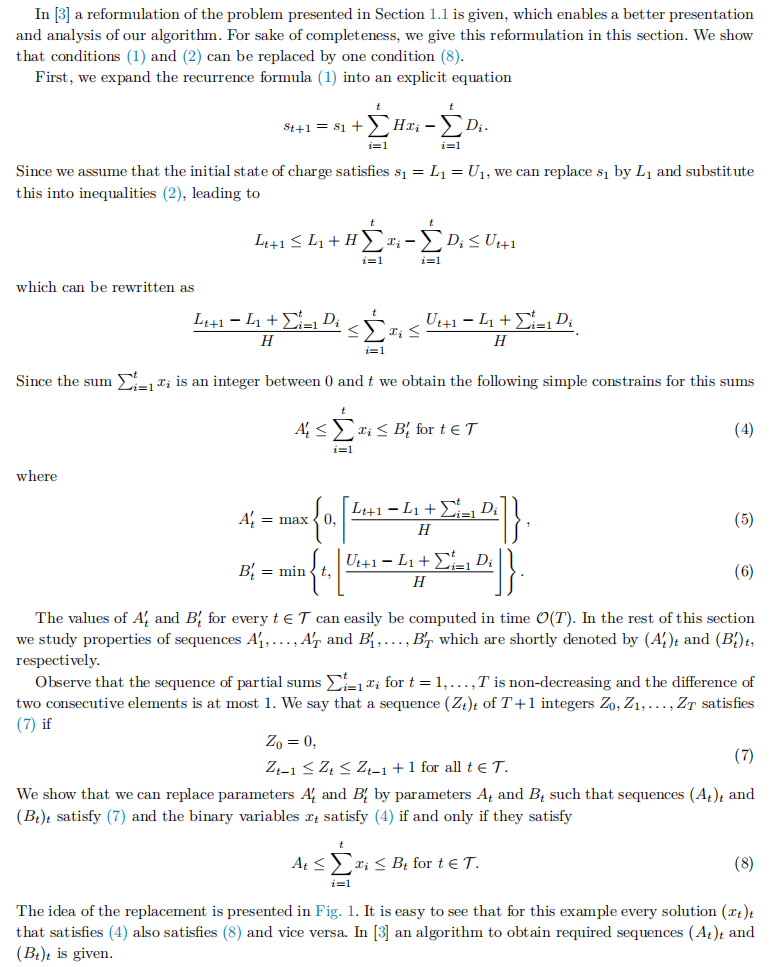
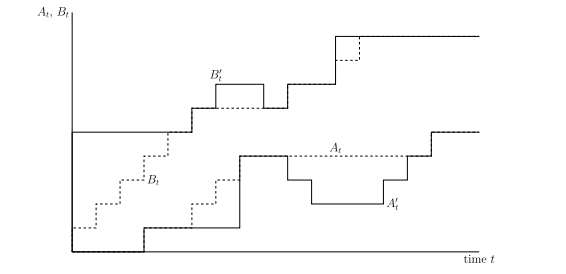
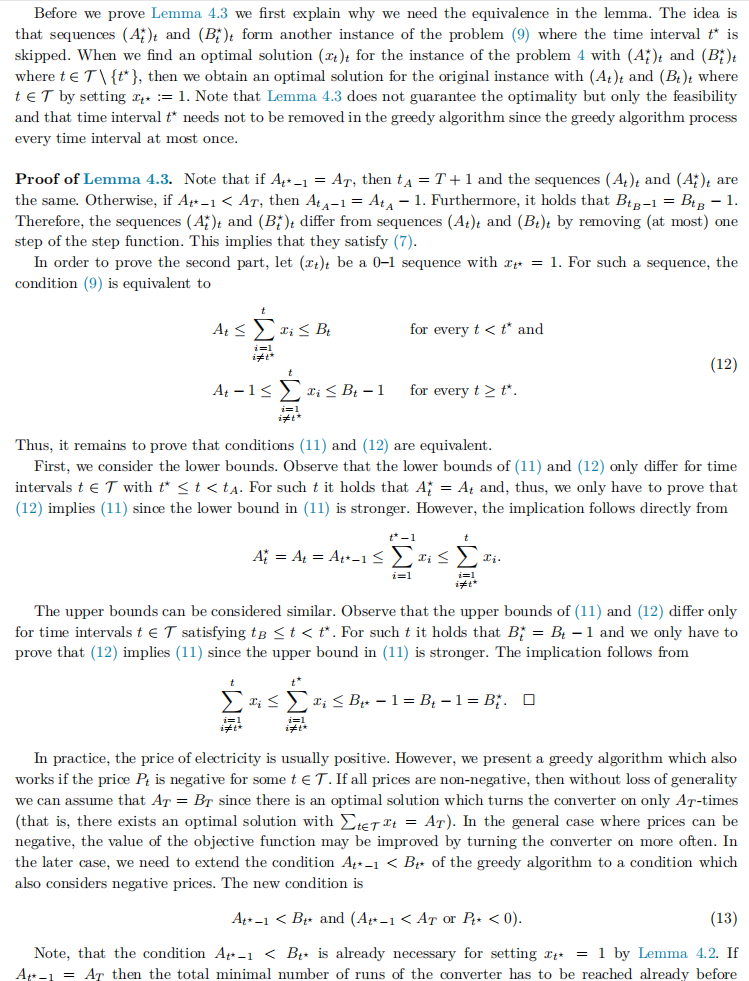
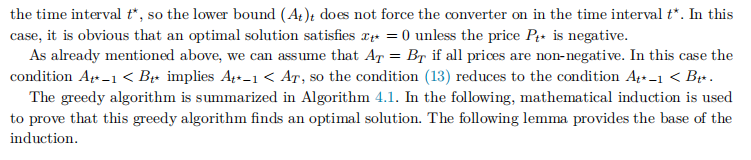
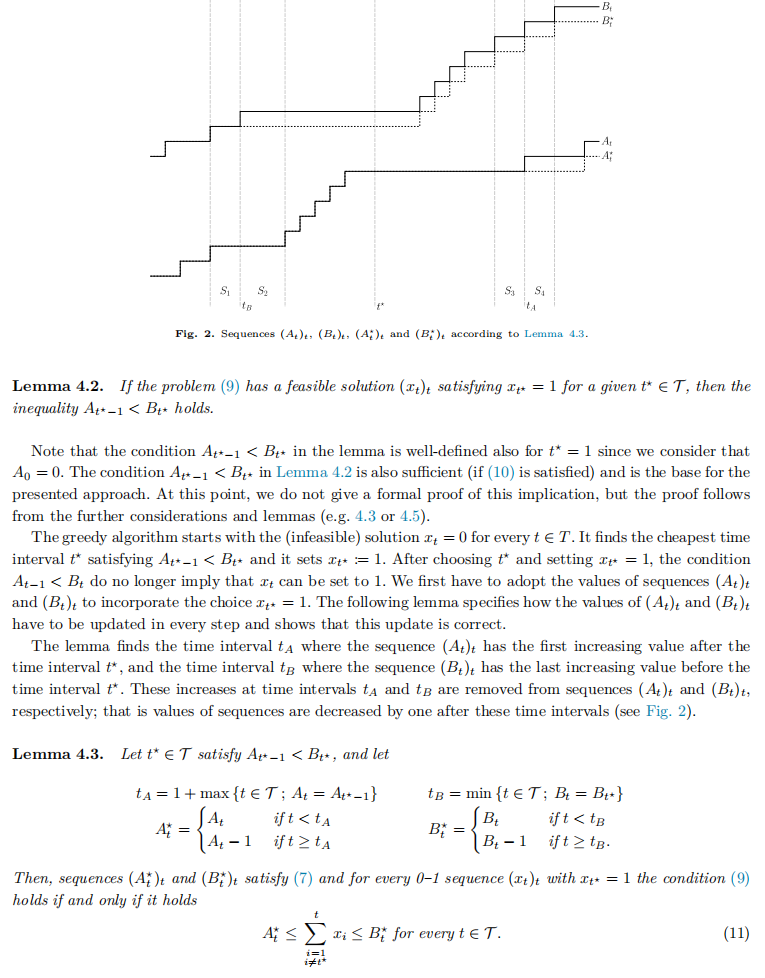
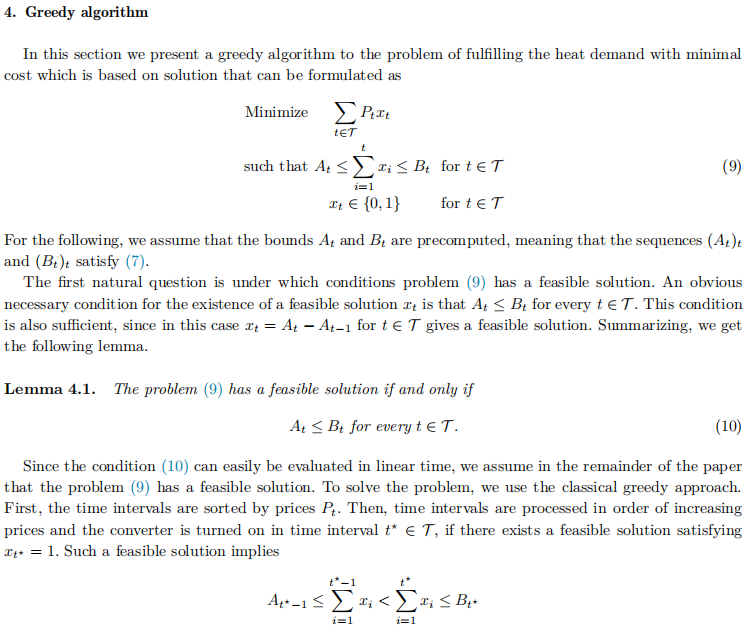
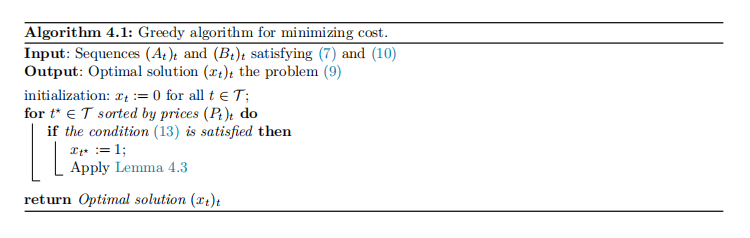
**Greedy algorithm for local heating problem**

**Description:**

**Algorithm Design and Proof of Correctness**





**Pseudo code**  ****

**Time Complexity**

In each loop, it needs to update the elements of A vector after ta and update the elements of B vector after tb, which takes O(T) on average, so the whole process takes O(T^2).

**Improvement**

Considering the need for ta in each update of the loop, the element after tb is subtracted by 1. Therefore, an array is maintained that allows the elements of an interval to be subtracted by a number, allowing the value of the subscript i to be interrogated. Using a line tree and using a difference array to update the values of the interval, assume that d[i] = a[i] - a[i-1] (d[0] = a[0]) , so subtracting a from the value in [l, r] requires only d[l] = d[l] - a, d[r + 1] = d[r + 1] - a, which requires O(1) operations. Applying the operation to the line tree, it takes O(log T) because each operation changes only one value at each level of the line tree. The query for the value of subscript i needs to consider the changes in the previous logi layer, and the query requires O(logT). In summary, the line-segment tree difference array can optimize the time complexity to O(TlogT).

1. The proof of correctness of segment tree

Theorem: A line tree of [1,n] can decompose any subinterval [L,R] of [1,n] into no more than IMG_256 subintervals for n >= 3.

Using mathematical induction, prove the above theorem as follows. First, for n=3,4,5, it is not difficult to prove the theorem by exhaustive enumeration. Suppose that for n= 3,4,5,... ,k-1 the above equation holds, the following is to prove that for n=k ( k>=6 ) holds.

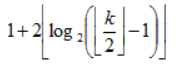
The proof is divided into 4 cases.

Case 1: [L,R] contains the root node (L=1 and R=n), at this time, [L,R] is decomposed into one node and the theorem holds.

Case 2: [L,R] contains the left child node of the root node, at this time [L,R] must not contain the right child node of the root (because if it does, it can merge the left and right child nodes and replace it with the root node, this is case one). At this point, the number of elements of the tree with the right child node as the root is .

The subinterval into which [L,R] is divided consists of two parts.

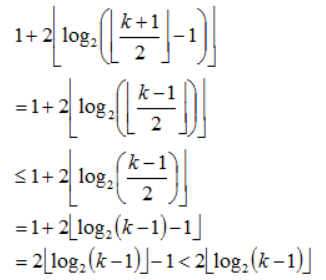
1. the left child node of the root, the number of intervals is 1
2. the interval query in the tree rooted by the right child node of the root, and this can be used recursively using this theorem.

From the induction hypothesis, it follows that [L,R] is divided into a total of  intervals.

Case 3: symmetric to case 2, unlike case 2, the number of elements of the tree rooted at the left child node of the root is IMG_256

.

The tree [L,R] is divided into a total of IMG_256 intervals. From the formula, we can see that the number of intervals in case two is less than or equal to the number of intervals in case three, so we only need to prove that the number of intervals in case three meets the condition.



Thus, the theorems of case two and case three hold.

Case 4: [L,R] does not include the root node and the left and right children of the root node. Thus, the remaining layer IMG_256 has at most two nodes per layer (refer to the content in the rough proof). Thus [L,R] is decomposed into at most IMG_256

intervals and the theorem holds. The above only proves that IMG_256is an upper bound, but, in fact, it is the minimum upper bound. When n=3,4, there are many groups of intervals that can be decomposed to reach the minimum upper bound. When n>4, the decomposition of the interval [L,R] can reach the minimum upper bound IMG_256when and only when n=2^t (t>=3),L=2,R=2^t - 1.

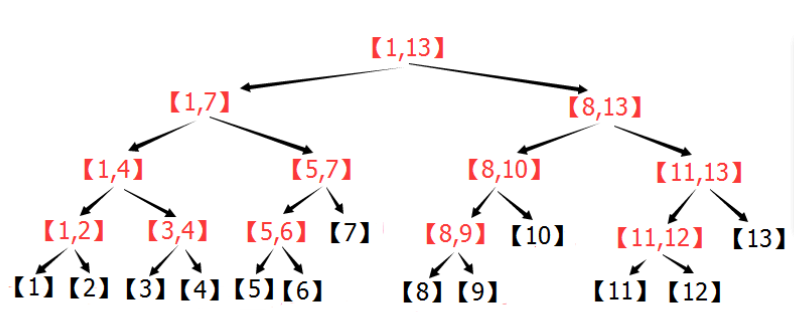
**Example**

A line tree is a decomposition of each interval [L,R] into [L,M] and [M+1,R] (where M=(L+R)/2 where the division is integer division, i.e., rounding down the result) until L==R.

The tree starts with the interval [1,n] , and is decomposed step by step by recursion, assuming that the height of the root is 1, and the maximum height of the tree is IMG_256(n>1).

The decomposition of the tree is unique for each n, so the tree structure is the same for the same n. This is also the basis for the implementation of a persistent tree.

The following figure shows the decomposition of the interval [1,13].



**Division of Labor**

韩子睿：program code

周德程：experimental report

孙宇祥：slides and presentation