RDARP is a variation of offline Dial-a-Ride, where each request has not only a source and destination but also a revenue that is earned for serving the request. The input to RDARP is a uniform metric space, a set of requests, a time limit T. Each request has a source point and a destination point in the metric space, and a revenue, where the revenues are nonuniform. A server starts at a designated point in the metric space, which is the origin. The goal is to move the server through the metric space, serving requests one at a time so as to maximize the revenue earned in T time units, with nonuniform revenues.

Algorithm 1 quickOPT2

Find the highest revenue set of requests S that can be completed in the next 2 time units
move to it
serve it

HRF' is a version of the Highest Revenue First algorithm, that operates only at even time units starting at $t = 0$. Thus it serves the highest revenue request available at the time units $t = 0, 2, 4, ..., T-1$.
OPT is an algorithm that yields an optimal result.

Algorithm 2 HRF'

if T is even **then**
    At evey even time, determine which request earns the greatest revenue and move to location of this request. Denote this request as r. If no unserved requests exist, do nothing until next even time.
else if T is odd **then**
    At time 0, do nothing.
    At every odd time, determine which request earns the greatest revenue and move to the source location of this request. Denote this request as r. If no unserved requests exist, do nothing until the next even time.
    At evey even time, complete request r from the previous step
end if

sortedOPT is a version of the OPT algorithm, that sorts all requests that OPT can serve by revenue in decreasing order: $r_1, r_2, \dots r_i$, where $r_1 >= r_2 >= \dots r_i$

Define rev(A) to be the revenue earned by the algorithm A.

The goal of this document is to prove the following theorems.

Theorem 1. *quickOPT2 is a 2-approximation for offline RDARP on the uniform metric.*

$$rev(quickOPT2) \geq rev(HRF') \geq \frac{1}{2} rev(sortedOPT) = \frac{1}{2} rev(OPT)$$

We will prove this theorem using the following lemmas.

Define what a window is. Let a window denote a pair of time units.

Let $rev(quickOPT2(w_1, ..., w_k))$ refer to the revenue earned by quickOPT2 in windows w_1 to w_k inclusive. Note $rev(HRF'(w_1, ..., w_k))$ is equal to the revenue of the top k highest-revenue requests.

Lemma 1. *For $k = 1...\frac{T}{2}$, $rev(quickOPT2(w_1, ..., w_k)) \geq rev(HRF'(w_1, ..., w_k))$*

Proof. **Base step**: $k = 1$. In w_1, HRF' will serve the request with $r_{\max}$. By definition, $quickOPT2$ will serve the highest revenue set of requests S that can be completed in the next 2 time unites. Therefore, $quickOPT2$ will either serve the request with the highest revenue $r_{\max}$ or the highest revenue set of request S such that $rev(S) \geq r_{\max}$. Hence, $rev(quickOPT2(w_1)) \geq rev(HRF'(w_1))$ holds for $k = 1$.

Inductive step: Assume $rev(quickOPT2(w_1, ..., w_k)) \geq rev(HRF'(w_1, ..., w_k))$ (ind. hyp.). We show that $rev(quickOPT2(w_1, ...w_{k+1})) \geq rev(HRF'(w_1, ..., w_{k+1}))$. There are two cases.

Case 1: $rev(quickOPT2(w_{k+1}) \geq rev(HRF'(w_{k+1}))$. Combining this inequality with the ind. hyp. implies $rev(quickOPT2(w_1, ...w_{k+1})) \geq rev(HRF'(w_1, ..., w_{k+1}))$

Case 2: $rev(quickOPT2(w_{k+1})) < rev(HRF'(w_{k+1}))$. Let v be the last request served by HRF' at w_{k+1}. By definition of $quickOPT2$, this must mean that $quickOPT2$ has already served v. (otherwise $quickOPT2$ would be doing v or better in w_{k+1}). Notice revenues earned by both algorithms per time window does not increase as the time window number increases. Formally, if h_i (and q_i, respectively) for $i = 1...T/2$ denotes the revenue earned by HRF' (and quickOPT2, respectively) in window i, then

$$h_i \geq h_j, \text{ for all } i, j \text{ where } i < j \text{ and } q_i \geq q_j, \text{ for all } i, j, \text{ where } i < j \tag{1}$$

Let w_j, for some $j < k + 1$, be the time window where $quickOPT2$ served v. Let Q_j be the set of served requests that $quickOPT2$ served up to and including those serve in w_j. Let H_{k+1} be the set of requests served by HRF' up to and including w_{k+1}. It follows that $H_{k+1} \subseteq Q_j$ by (Δ). Thus, $rev(H_{k+1}) \leq rev(Q_j)$.

Δ: Suppose for contradiction that there is some request v' in H_{k+1} not in Q_j. Note $v' > v$ since it is served before v in the HRF' schedule (since v was served in w_{k+1}). But $quickOPT2$ served v in w_j instead of v', while v' was still available to be served by quickOPT2. This contradicts the definition of $quickOPT2$.

Hence,

$$rev(HRF'(w_1, ..., w_{k+1})) \leq rev(quickOPT2(w_1, w_2, ..., w_j)) \leq rev(quickOPT2(w_1, ..., w_{k+1}))$$

Therefore, the addition of the revenue earned by HRF' from v in w_{k+1} will not be enough to make $HRF'(w_1....w_{k+1}) > quickOPT2(w_1...w_{k+1})$ by (4). □

Lemma 2. $rev(HRF') \geq \frac{1}{2}rev(sortedOPT) = \frac{1}{2}rev(OPT)$.

Proof. Let the sequence of $sortedOPT$ requests be denoted as $r_1, r_2, ..., r_n$, where $n \leq T$. Then,

$$\sum_{i=1}^{T/2} r_i \geq \frac{1}{2}\sum_{i=1}^{n} r_i = \frac{1}{2}rev(sortedOPT) = \frac{1}{2}rev(OPT) \tag{2}$$

because $r_i \geq r_j$ for $i < j$, and $n \leq T$.

Denote the sequence of HRF' requests as $h_1, h_2, ..., h_{T/2}$. By greediness of HRF', $h_i \geq r_i$ for all $1 \leq i \leq T/2$. Hence,

$$rev(HRF') = \sum_{i=1}^{T/2} h_i \geq \sum_{i=1}^{T/2} r_i \tag{3}$$

By equations (2) and (3), we have shown $rev(HRF') \geq \frac{1}{2}rev(sortedOPT) = \frac{1}{2}rev(OPT)$. □

Theorem 2. *quickOPT3 is a $\frac{2}{3}$-approximation for offline RDARP on the uniform metric.*

$$rev(quickOPT3) \geq rev(HR2F) \geq \frac{2}{3}OPT$$

We will prove this theorem using the following lemmas.

Define what a window is. Let a window denote 3 time units. Assume for now that $T = 3k$.

Let $rev(quickOPT3(w_1, ..., w_k))$ refer to the revenue earned by quickOPT3 in windows w_1 to w_k inclusive. $HR2F$ is the revenue version of two-chain algorithm.

Algorithm 3 HR2f

```
Input: Set S of requests, time limit T, origin
Initialize server to origin
while available requests remain and time has not run out do
    if any 2-chains remain then
        find the highest-revenue 2-chain with revenue max(rev((u, v)))
    else
        Find the highest revenue 1-chain (singleton) with revenue max(rev(w))
    end if
    Serve either (u, v) or w, the one with higher-revenue.
end while
```

Lemma 3. *For $k = 1...\frac{T}{2}, rev(quickOPT3(w_1, ..., w_k)) \geq rev(HR2F(w_1, ..., w_k))$.*

Proof. **Base step**: $k = 1$. In w_1, $HR2F$ will serve a chain of 2 requests with the total revenue $r_{\max}$. By definition, $quickOPT3$ will serve the highest revenue set of requests S that can be completed in the next 3 time units. Therefore, $quickOPT3$ will either serve the two-chain with the highest total revenue $r_{\max}$ or the highest revenue set of request S such that $rev(S) \geq r_{\max}$. Hence, $rev(quickOPT3(w_1)) \geq rev(HRF'(w_1))$ holds for $k = 1$.

Inductive step: Assume $rev(quickOPT3(w_1, ..., w_k)) \geq rev(HR2F(w_1, ..., w_k))$ (ind. hyp.). We show that $rev(quickOPT3(w_1, ...w_{k+1})) \geq rev(HR2F(w_1, ..., w_{k+1}))$. There are two cases.

Case 1: $rev(quickOPT3(w_{k+1}) \geq rev(HR2F(w_{k+1}))$. Combining this inequality with the ind. hyp. implies $rev(quickOPT3(w_1, ...w_{k+1})) \geq rev(HR2F(w_1, ..., w_{k+1}))$

Case 2: $rev(quickOPT3(w_{k+1})) < rev(HR2F(w_{k+1}))$. Let u, v be the last two-chain served by HR2F at w_{k+1}. By definition of $quickOPT3$, this must mean that $quickOPT3$ has already served at least one of u or v (or both). (Otherwise $quickOPT3$ would be serving u, v or some higher-revenue set in w_{k+1}). Notice revenues earned by both algorithms per time window decreases as the time window number increases. Formally, if h_i (and q_i, respectively) for $i = 1...T/2$ denotes the revenue earned by HR2F (and quickOPT3, respectively) in window i, then

$$h_i \geq h_j, \text{ for all } i, j \text{ where } i < j \text{ and } q_i \geq q_j, \text{ for all } i, j, \text{ where } i < j \tag{4}$$

Let w_j, for some $j < k+1$, be the time window where $quickOPT3$ served v. Let Q_j be the set of served requests that $quickOPT3$ served up to and including those serve in w_j. Let H_{k+1} be the set of requests served by $HR2F$ up to and including w_{k+1}. It follows that $H_{k+1} \subseteq Q_j$ by (Δ). Thus, $rev(H_{k+1}) \leq rev(Q_j)$.

Δ: Suppose for contradiction that there is some request v' in H_{k+1} not in Q_j. Note $v' > v$ since it is served before v in the $HR2F$ schedule (since v was served in w_{k+1}). But $quickOPT3$ served v in w_j instead of v', while v' was still available to be served by quickOPT3. This contradicts the definition of $quickOPT3$.

Hence,

$$rev(HRF'(w_1, ..., w_{k+1})) \leq rev(quickOPT3(w_1, w_2, ..., w_j)) \leq rev(quickOPT3(w_1, ..., w_{k+1}))$$

Therefore, the addition of the revenue earned by HRF' from v in w_{k+1} will not be enough to make $HRF'(w_1....w_{k+1}) > quickOPT3(w_1...w_{k+1})$ by (4). □