

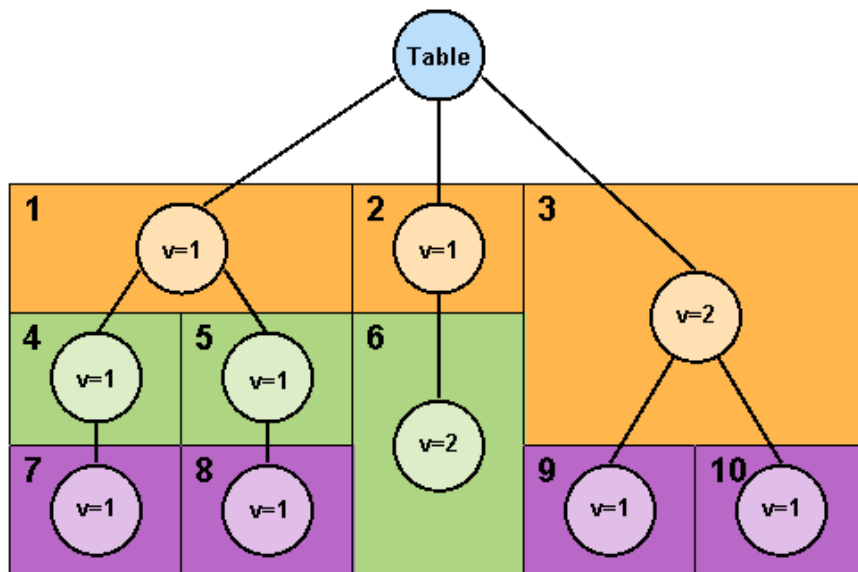
Вставка строк и колонок в таблицу

Описание задачи

Условие

Дано: таблица, структура которой представима в виде «дерева». (Другие таблицы [некорректны](#).)

Ниже иллюстрация представимости структуры таблицы в виде «дерева» и пример JSON:



```
{
  "Name": "Table",
  "Children": [
    {
      "VerticalSpan": 1, "Color": "Orange", "Value": "1",
      "Children": [
        {
          "VerticalSpan": 1, "Color": "Green", "Value": "4",
          "Children": [ { "VerticalSpan": 1, "Color": "Purple", "Value": "7", "Children": [] } ]
        },
        {
          "VerticalSpan": 1, "Color": "Green", "Value": "5",
          "Children": [ { "VerticalSpan": 1, "Color": "Purple", "Value": "8", "Children": [] } ]
        }
      ]
    },
    {
      "VerticalSpan": 1, "Color": "Orange", "Value": "2",
      "Children": [ { "VerticalSpan": 2, "Color": "Green", "Value": "6", "Children": [] } ]
    },
    {
      "VerticalSpan": 2, "Color": "Orange", "Value": "3",
      "Children": [
        { "VerticalSpan": 1, "Color": "Purple", "Value": "9", "Children": [] },
        { "VerticalSpan": 1, "Color": "Purple", "Value": "10", "Children": [] }
      ]
    }
  ]
}
```

Постановка

Необходимо реализовать вставку строк и колонок в таблицу в указанной позиции. (Для примера можете посмотреть Microsoft Excel или аналогичные приложения.)

Следует обратить внимание на то, что некоторые ячейки таблицы могут [объединёнными](#) (иметь неединичные [спаны](#)). Соответственно, если строка (колонок) вставляется так, что пересекает ячейку с неединичным вертикальным (горизонтальным) спаном, такая ситуация должна быть корректно обработана (то есть, спан должен быть увеличен).

Требования

1. При реализации нужно воспользоваться структурой данных «дерево».
2. Исходная таблица не должна модифицироваться при вставке новых строк/колонок. Однако при этом не следует пользоваться ни стандартными, ни сторонними функциями, просто создающими «глубокую копию» исходной таблицы. [Ниже](#) представлена функция, которую необходимо использовать в данной задаче.

Визуализация

Для визуальной оценки реализации необходимо подготовить тестовые данные – несколько исходных таблиц. Для этих таблиц Вам нужно написать набор тестов и отобразить результат:

(исходная таблица, позиция для вставки строки/колонки) => (результатирующая таблица)

Тесты должны быть показательными, включающими в себя как простые, так и сложные случаи.

Для визуализации воспользуйтесь библиотекой [React](#) или другими удобными Вам средствами.

Этапы реализации

Вы можете заниматься реализацией задачи поэтапно – от простого к сложному:

0. Базовая реализация – вставка строк и колонок в таблицу, состоящую только из [единичных ячеек](#), то есть спанов в таблице нет совсем.
1. Далее реализуйте поддержку для таблиц, имеющих в структуре только один вид спанов (или горизонтальные, или вертикальные).
2. Следующим шагом добавьте работу с таблицами, содержащими оба вида спанов (и горизонтальные, и вертикальные).

Даже если с каким-то этапом возникнут трудности, присылайте реализованные этапы.

Дополнительное задание

Решение этого задания будет отдельным плюсом.

Мы предлагаем Вам реализовать undo-redo для операций над таблицей. Тот факт, что исходная таблица при вставке строк и колонок не должна модифицироваться, очень поможет при написании undo-redo.

Каким образом продемонстрировать работу undo-redo, мы оставим на Ваше усмотрение.

Инструментарий

0. При реализации обязательной к использованию является функция **update()** (<https://facebook.github.io/react/docs/update.html>) из пакета **react-addons-update** (<https://www.npmjs.com/package/react-addons-update>). Именно эта функция должна использоваться для получения результирующей таблицы со вставленными строками/колонками.
1. В качестве среды мы предлагаем воспользоваться JSFiddle (<https://jsfiddle.net/>). Для работы на этом портале нужно зарегистрироваться. Также нами был подготовлен пример (<https://jsfiddle.net/pvrte8zb/>), где уже добавлены зависимости на библиотеку **React** (<https://facebook.github.io/react/>) и функцию **update()**.
2. Допускается добавлять зависимости и на другие библиотеки, если их использование не превращает реализацию задачи в «тривиальную» – «однострочный» вызов одной или нескольких сторонних функций ☺

Документация

Результатом выполнения задачи является не только реализация, но и сопутствующая документация, включающая следующие пункты:

- Инструкция

Необходимо пояснить, как запустить реализацию (ссылка на JSFiddle или другую среду, где делалась задача). Если для запуска требуются ещё какие-то шаги, указать их.

- Ограничения реализации

Может оказаться так, что Ваша реализация будет покрывать не все возможные случаи – тогда необходимо описать, при каких входных данных реализация не будет корректно работать.

- Как можно улучшить реализацию

Вероятно, у Вас будут идеи, каким образом текущая реализация может быть улучшена (оптимизирована по производительности, поправлены ограничения в реализации и т.д.)

Глоссарий

Единичная ячейка

Ячейка называется единичной, если она не была получена путём объединения нескольких ячеек.

Некорректная таблица

Таблица некорректна, если её структура не представима в виде «дерева», например:

Wrong!		

Такие таблицы в данной задаче не рассматриваются.

Объединённая ячейка

Ячейка таблицы называется объединённой, если она получена путём объединения нескольких ячеек. Ячейки могут быть объединены, если они являются соседними и результат их объединения имеет прямоугольную форму. (В целом, как в Microsoft Excel или аналогичных приложениях.)

Спан

Спан – от английского span (диапазон, интервал, промежуток) – это свойство ячейки, он может быть вертикальным и горизонтальным.

Если ячейка единичная, то её вертикальный и горизонтальный спаны равны единице.

Для объединённой ячейки значения спанов соответствуют количеству единичных ячеек по вертикали и горизонтали, из которых была получена эта объединённая ячейка.