Universidad de Talca
Facultad de Ingeniería
Departamento de Ciencias de la Computación

Diseño y análisis de algoritmos

Cálculo de mediana en tiempo lineal como mejor caso y caso promedio y estabilización del algoritmo de selección por pivotes IQS
Informe 2

Erik Regla
eregla09@alumnos.utalca.cl

7 de Julio del 2015

1. Descripción de las pruebas experimentales

Para cada uno de los algoritmos desarrollados, se utilizó un conjunto de pruebas sintético por décadas con un tamaño total de $\sim$ 11GB. Cada prueba fue realizada con un conjunto de elementos bajo una distribución uniforme con un 5 % máximo de elementos repetidos. Originalmente este trabajo incluía el estudio del efecto de estos algoritmos sobre el mecanismo de selección incremental IQS, pero esto ha sido reemplazado por el estudio de una propuesta introspectiva al algoritmo de selección de medianas en tiempo lineal que incremente la precisión junto con su estabilidad en el caso promedio y evaluar sus efectos sobre el peor caso sin afectar su orden.

1.1. Leyenda en gráficos

- Violeta: Algoritmo de selección de mediana en tiempo $O(n)$ en caso promedio (QuickSelect).
- Café: Algoritmo de selección de mediana en tiempo $O(n)$ en peor caso (BFPRT) con una iteración sobre el arreglo y luego mediana de medianas de un nivel.
- Azul: Algoritmo de selección de mediana en tiempo $O(n)$ en peor caso (iBFPRT) iterado sobre el resultado.
- Rojo: Algoritmo de selección de mediana introspectivo en tiempo $O(n)$ en peor caso (IntrospectiveQuickMedian).

1.2. QuickSelect

Como es posible observar, si bien el rendimiento en el caso promedio es bastante prometedor y describe un comportamiento mas menos lineal (Figura 3), en el peor caso (Figuras 2 y 3) muestra

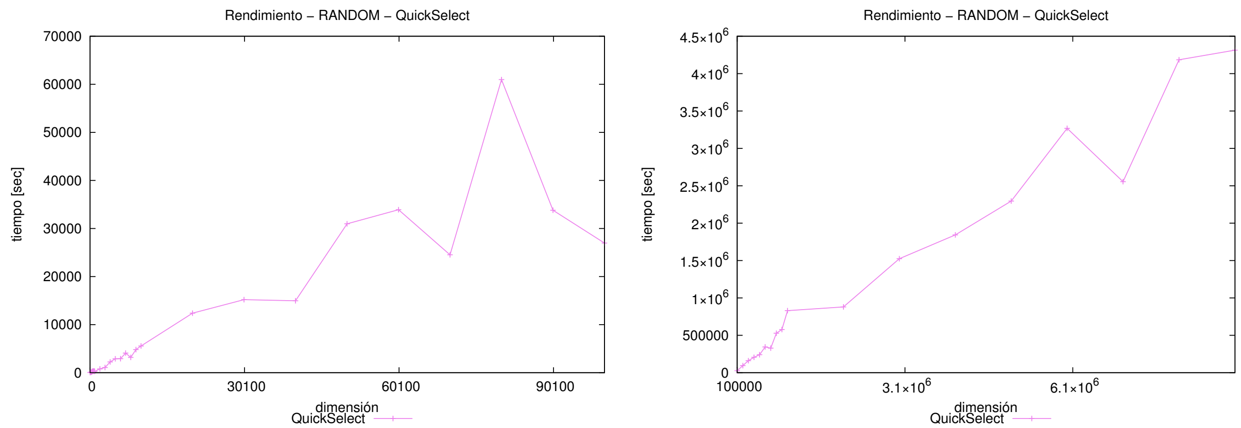


Figura 1: Rendimiento de QuickSelect para encontrar la mediana de un conjunto desordenado.

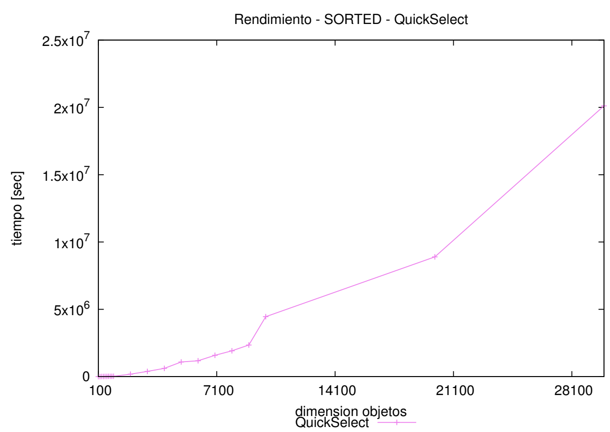


Figura 2: Rendimiento de QuickSelect para encontrar la mediana de un conjunto ordenado.

claramente un comportamiento cuadrático. Debido a esto, los experimentos luego de $N = 30000$ fueron inviables por tiempo. La precisión de este algoritmo en encontrar la mediana es de 100 %

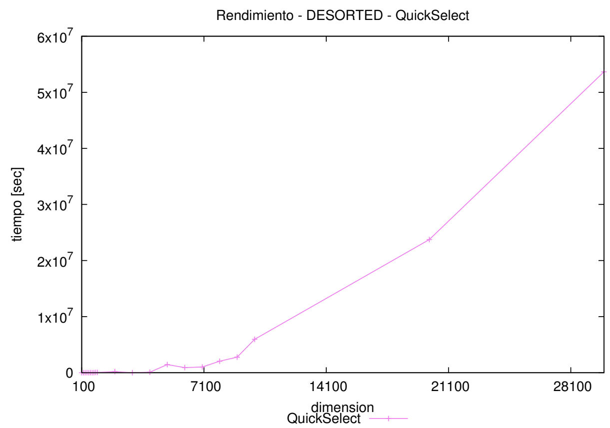
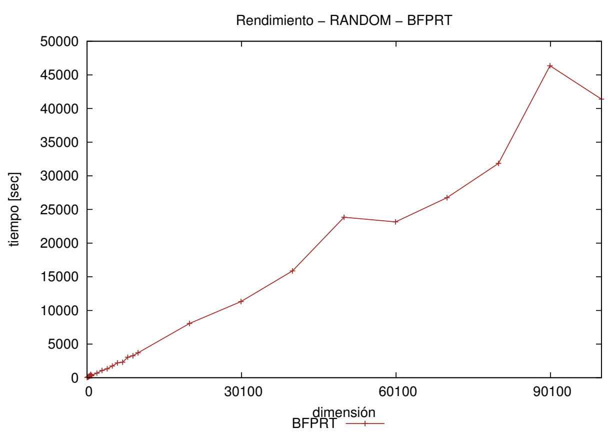


Figura 3: Rendimiento de QuickSelect para encontrar la mediana de un conjunto ordenado inversamente.



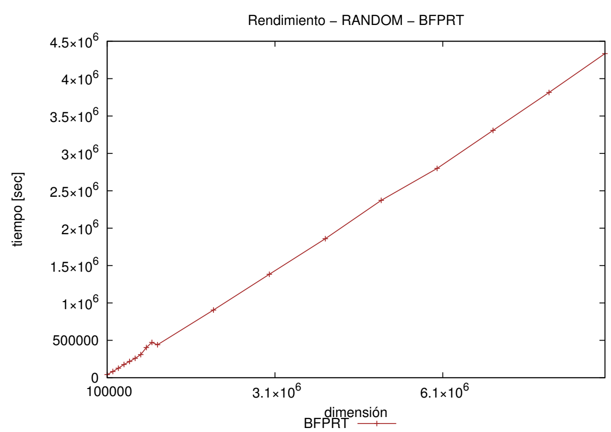


Figura 4: Rendimiento de BFPRT no iterado para encontrar la mediana de un conjunto desordenado.

1.3. Mediana de medianas no iterada

Para el caso de la selección utilizando una mediana no iterada, se ve un rendimiento similar a de un algoritmo de orden $n\ lg(n)$, principalmente debido a que dado que la mediana de medianas no es iterada, luego de obtener el primer conjunto de candidatos requiere ordenarlos para poder seleccionar el elemento medio. Utilizar BFPRT de este modo, no es diferente a utilizar el método de *mediana de 3* solo que con $\frac{n}{5}$ elementos. Esto no es apreciable a simple vista, dado que al dividir por 5 el valor de $lg(n)$ este baja bastante su valor, además de tener una precisión bastante alta, fluctuando entre un 98 % y 100 % en todos los casos.

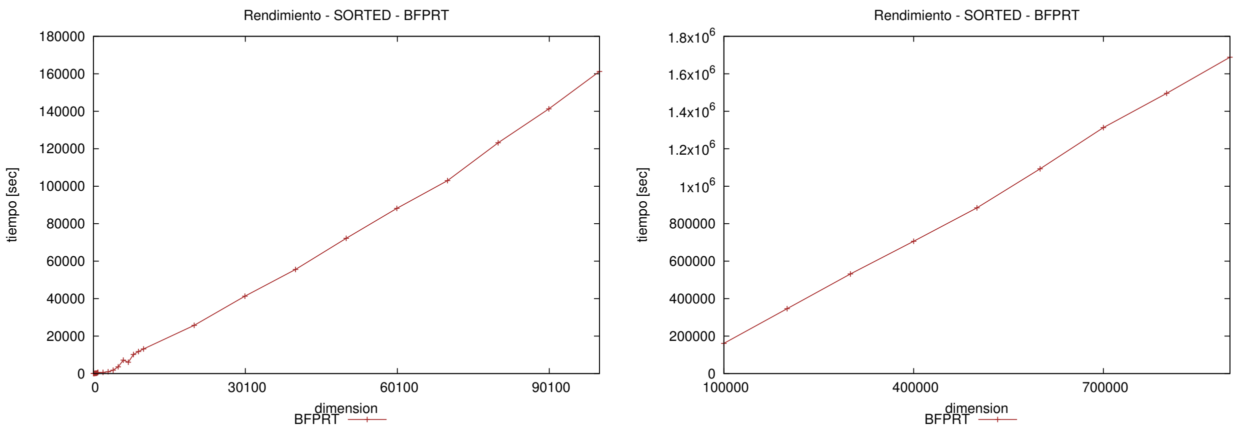


Figura 5: Rendimiento de BFPRT no iterado para encontrar la mediana de un conjunto ordenado.

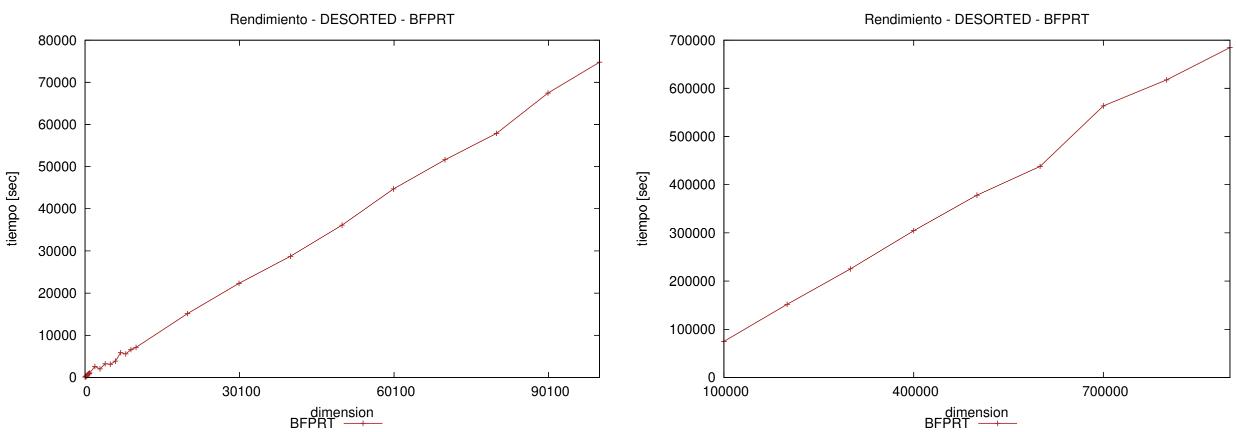


Figura 6: Rendimiento de BFPRT no iterado para encontrar la mediana de un conjunto ordenado inversamente.

1.4. Mediana de medianas iterada versus versión no iterada

Es posible observar un incremento del rendimiento en comparación a su par no iterado, principalmente debido a que la complejidad baja en $\frac{5}{n}$ en cada iteración y la operación de selección de mediana se ejecuta bajo un tiempo constante, asegurando una ejecución en peor caso de tiempo lineal. Existe una significativa pérdida de precisión respecto a su par no iterado, principalmente debido a que pierde información en cada ciclo. Su precisión fluctúa entre el 94 % a 99 % para el caso promedio y entre 84 % a 99 % para el peor caso.

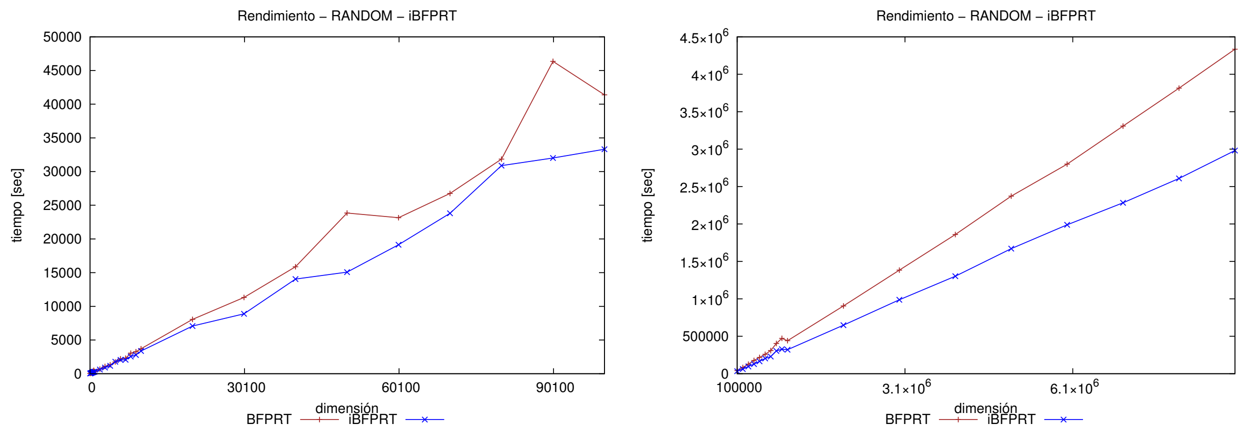


Figura 7: Rendimiento de iBFPRT iterado para encontrar la mediana de un conjunto desordenado.

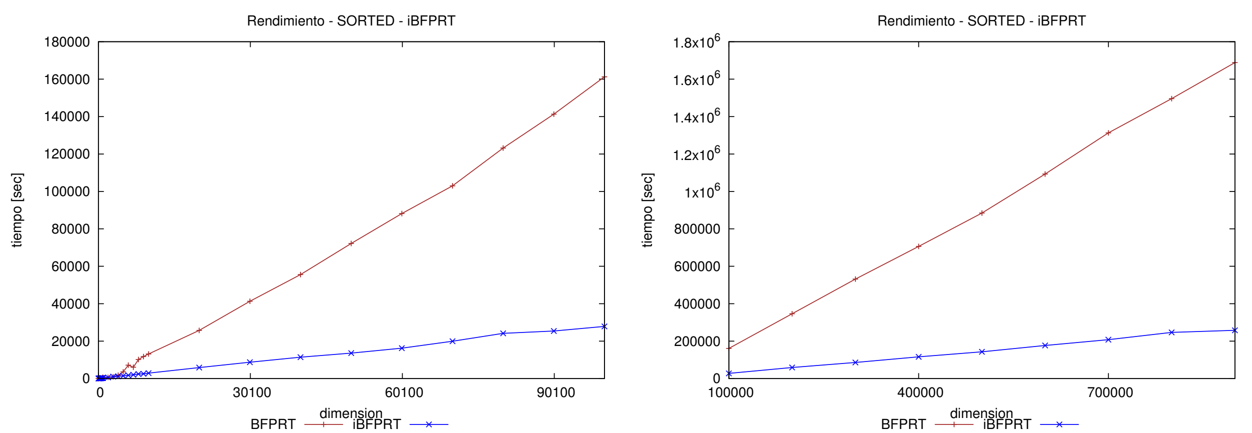


Figura 8: Rendimiento de BFPRT iterado para encontrar la mediana de un conjunto ordenado.

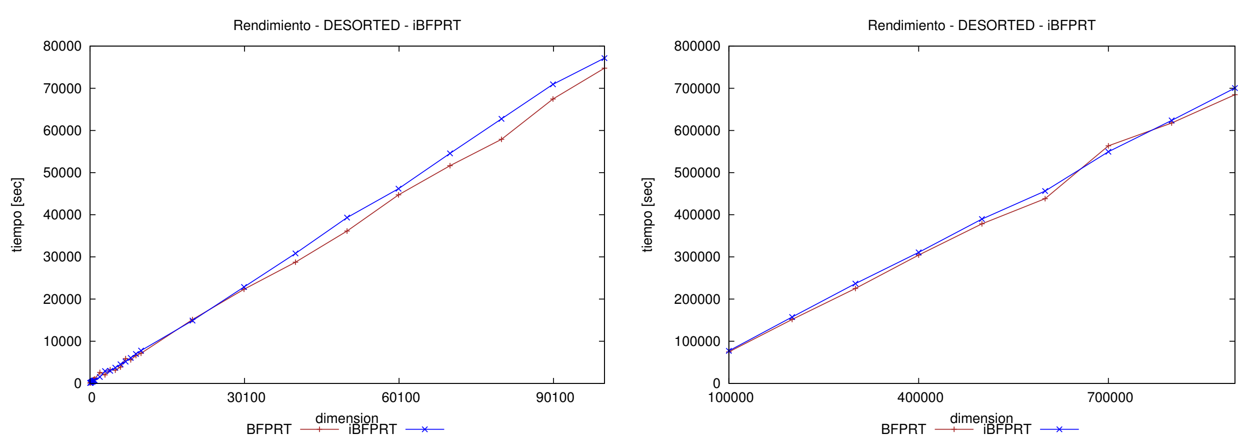


Figura 9: Rendimiento de BFPRT iterado para encontrar la mediana de un conjunto ordenado inversamente.

1.5. Selección introspectiva de mediana aproximada

Esta modificación del algoritmo BFPRT utiliza como base la implementación existente de QuickSelect. Dado que en el caso promedio QuickSelect realiza $lg(n)$ recursiones para encontrar el valor buscado, es posible afirmar que si la cantidad de recursiones es superior a este valor, entonces estamos cayendo en el peor caso, y entonces cambia a BFPRT para buscar la mediana aproximada. La ventaja de este algoritmo es que en el caso promedio posee una precisión más alta que el algoritmo de mediana de medianas compitiendo incluso contra QuickSelect y en el peor caso mantiene o mejora la precisión del algoritmo de mediana de medianas. Su precisión fluctúa entre 99 % a 100 % para el caso promedio y 85 % a 99 % en el peor caso.

Algoritmo 1 Selección de mediana aproximada utilizando introspección

Precondición: J arreglo de datos desordenados, k el largo de la lista temporal K, i, j son los límites del arreglo J

Postcondición: $A_{\frac{|A|}{2}}$ es la mediana aproximada de J

si recursión en peor caso **entonces**
    $A \leftarrow []$
    $i \leftarrow 0$
    mientras $i + k < |A|$ **hacer**
        $K \leftarrow A_{i...min(i+k,|A|)}$
        $K \leftarrow ordenar(K)$
        $insertar K_{\frac{|K|}{2} en A}$
        $i \leftarrow i + k$
    fin mientras
    $A \leftarrow ordenar(A)$
    devolver $medianaDeMedianas(A)$
si no
    $i \leftarrow 0$
    $j \leftarrow |A|$
    si $p \neq \frac{|A|}{2}$ **entonces**
        $p \leftarrow obtenerPivote(A, i, j)$
        $p \leftarrow particionar(A, i, j)$
        si $p > \frac{|A|}{2}$ **entonces**
            $i \leftarrow p$
        si no
            $j \leftarrow p$
        fin si
        devolver $IntrospectiveQuickMedian(A, i, j)$
    si no
        devolver p
    fin si
fin si

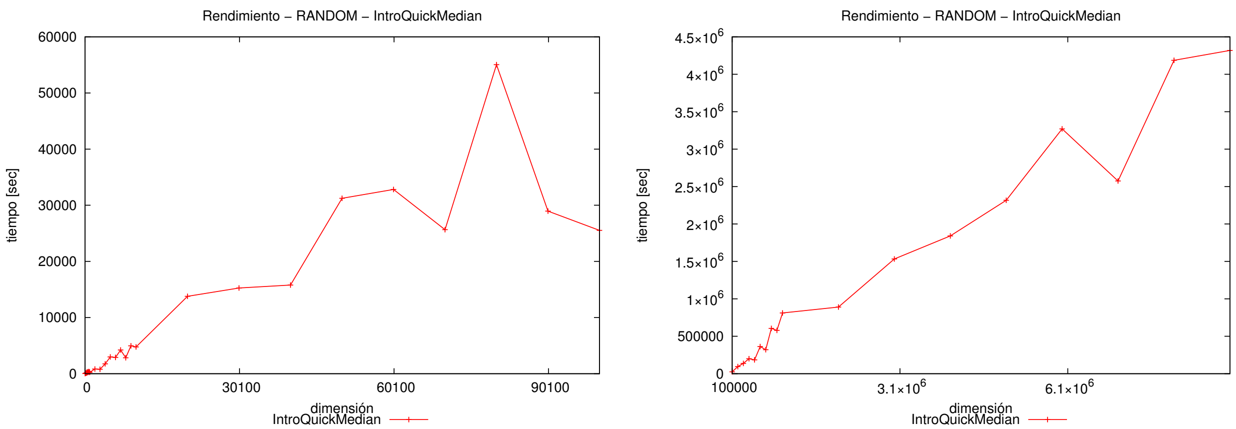


Figura 10: Rendimiento de IQM iterado para encontrar la mediana de un conjunto desordenado.

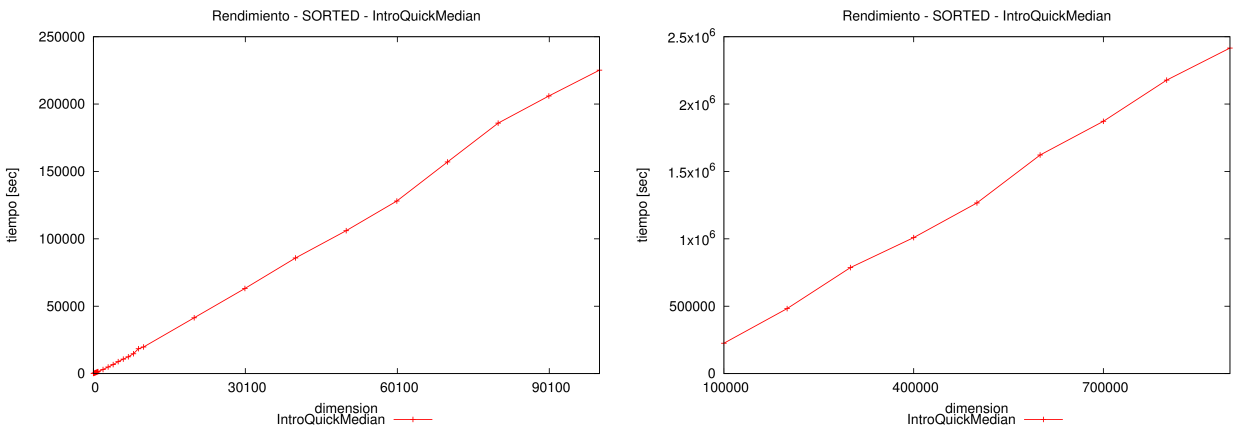


Figura 11: Rendimiento de IQM iterado para encontrar la mediana de un conjunto ordenado.

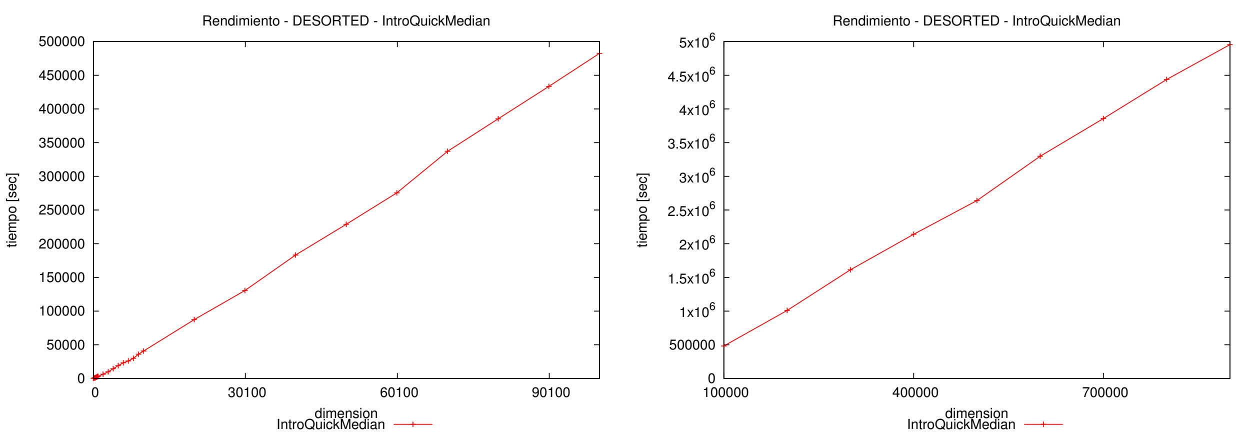


Figura 12: Rendimiento de IQM iterado para encontrar la mediana de un conjunto ordenado inversamente.

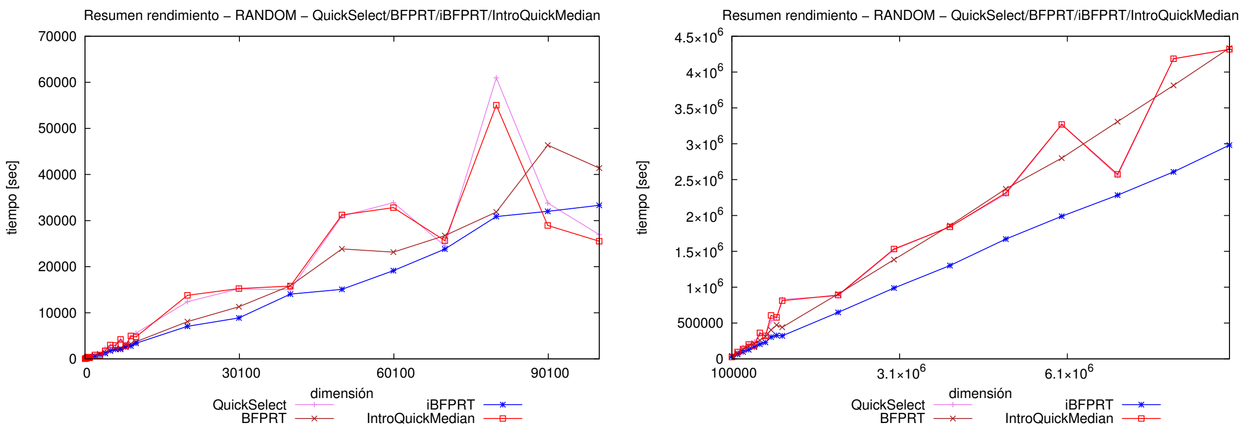


Figura 13: Rendimiento de algoritmos para selección de mediana para encontrar la mediana de un conjunto desordenado.

2. Resumen y comparación final

Finalmente, comparando los algoritmos en estudio, se puede apreciar que el mejor rendimiento y estabilidad para el caso promedio es presentado por el algoritmo BFPRT iterado, seguido de QuickSelect e IntrospectiveQuickMedian, y para el peor caso la mejor estabilidad y rendimiento es evidente por la familia de algoritmos basados en mediana de medianas seguido por IntrospectiveQuickSelect con un orden aparentemente similar, en cambio QuickSelect muestra un comportamiento erratico con un consumo de tiempo excesivo, propio del peor caso esperado.

Respecto a la precisión alcanzada para el caso promedio QuickSelect presenta una precisión absoluta de 100 % seguida muy de cerca por IntrospectiveQuickMedian y luego BFPRT en sus versiones no iteradas e iteradas siendo esta última con la menor precisión, y para el peor caso, la mejor precisión es alcanzada por QuickSelect y BFPRT no iterado y luego BFPRT iterado junto con IntrospectiveQuickMedian, sin una pérdida significativa de precisión respecto a BFPRT iterado.

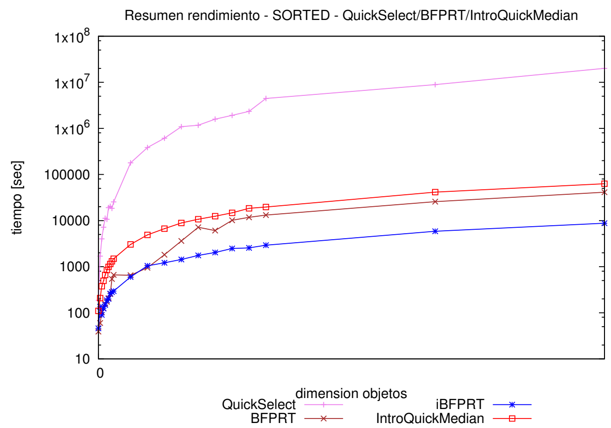


Figura 14: Rendimiento de algoritmos de selección de medianas para encontrar la mediana de un conjunto ordenado.

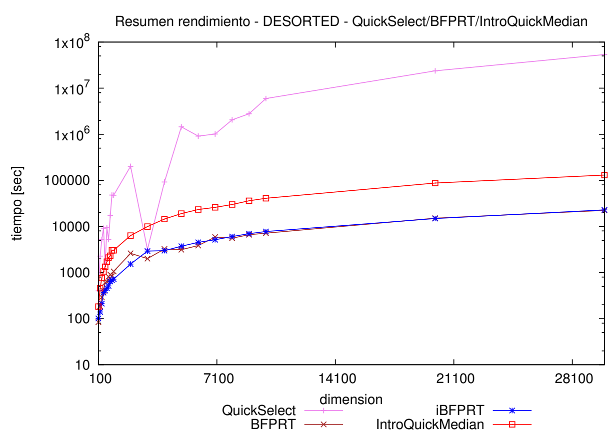


Figura 15: Rendimiento de algoritmos para selección de medianas para encontrar la mediana de un conjunto ordenado inversamente.

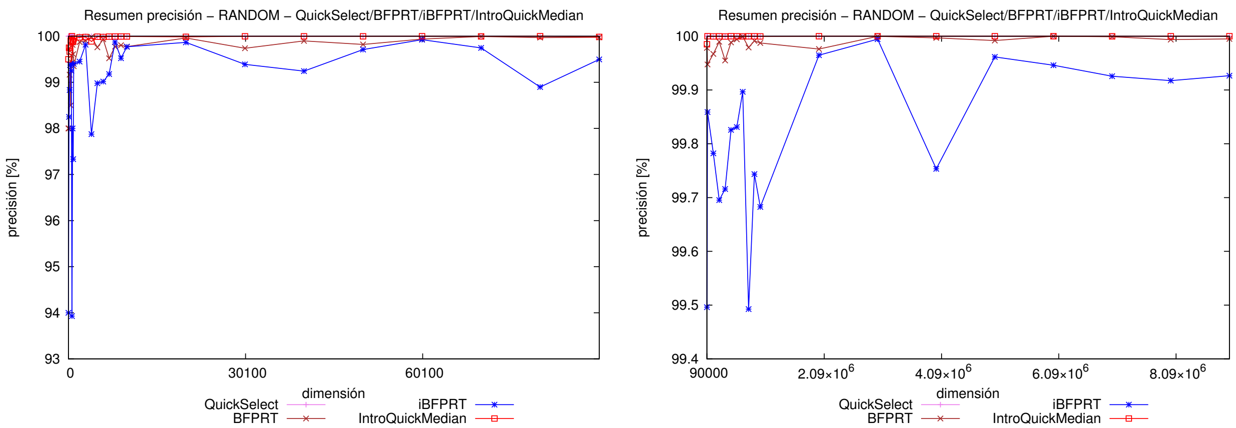


Figura 16: Precisión de algoritmos para selección de mediana para encontrar la mediana de un conjunto desordenado.

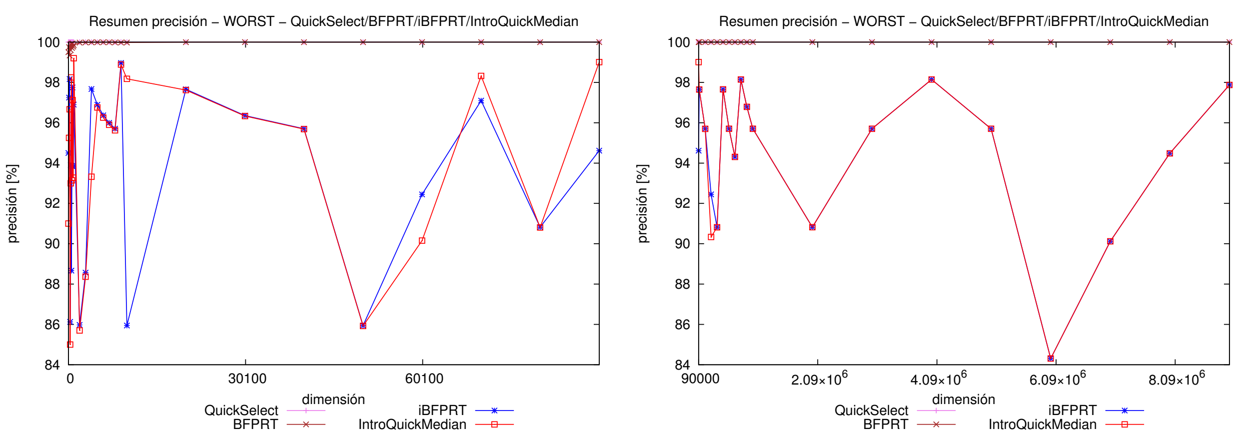


Figura 17: Precisión de algoritmos para selección de mediana para encontrar la mediana de un conjunto ordenado inversamente.

3. Trabajo Futuro

- Investigar el efecto que produce reducir el espacio muestral utilizando la técnica empleada en BFPRT no iterado
- Investigar el efecto que tiene este algoritmo de selección de medianas en algoritmos de selección basados en pivotes.
- Investigar el efecto que tiene el criterio de selección de espacio muestral sobre la condición de introspección.